



**Sérgio Miguel da Silva Barata Onofre**

Mestre em Engenharia Eletrotécnica e de Computadores

## **Arquitetura baseada em multiagentes georreferenciados para Sistemas de Vigilância**

Dissertação para obtenção do Grau de Doutor em  
Engenharia Eletrotécnica e de Computadores

Orientador: Doutor Pedro Alexandre da Costa Sousa,  
Professor Auxiliar, FCT-UNL

Co-orientadores: Doutor João Paulo Branquinho Pimentão,  
Professor Auxiliar, FCT-UNL

Júri:

Presidente: Doutor Jorge Joaquim Pamiés Teixeira

Arguentes: Doutora Ernestina Menasalvas Ruiz  
Doutor José António Barata de Oliveira

Vogais: Doutor Massimiliano Zanin  
Doutor Jorge Joaquim Pamiés Teixeira  
Doutor João Paulo Branquinho Pimentão



## **Arquitetura baseada em multiagentes georreferenciados para Sistemas de Vigilância**

Copyright © Sérgio Miguel da Silva Barata Onofre, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



*Dedico esta dissertação às minhas meninas.*



## Agradecimentos

Este espaço é dedicado àqueles que deram a sua contribuição para que esta dissertação fosse realizada. A todos eles deixo aqui o meu agradecimento sincero.

Agradeço aos meus orientadores Professor Doutor Pedro Sousa e Professor Doutor João Paulo Pimentão pela forma como guiaram o meu trabalho e por todas as recomendações que me foram dando ao longo do mesmo, permitindo que este contribuísse para o meu desenvolvimento pessoal. Quero também agradecer aos elementos integrantes da minha CAT (Comissão de Acompanhamento de Tese), a Professora Doutora Ernestina Menasalvas e ao Professor Doutor José Barata, pelos conselhos dados durante a supervisão da execução do meu trabalho.

Deixo também aqui uma nota de apreço aos meus amigos e colegas de trabalho da HOLOS, pelo bom ambiente e partilha de conhecimento prestada e sobretudo pela amizade e entajuda que sempre me têm demonstrado.

Gostaria de deixar aqui o meu agradecimento muito especial à minha mulher Rita pelo apoio e compreensão nos momentos mais difíceis e pelo encorajamento que sempre me transmitiu durante a realização desta dissertação e agradecer aos meus pais e irmã pelo apoio que sempre têm demonstrado ao longo da minha vida, incentivando-me sempre a atingir novas metas.

Por último, quero deixar o meu agradecimento à HOLOS que me permitiu participar nos projetos DVA e ServRobot, onde parte do meu trabalho foi desenvolvido, e pelo apoio institucional que sempre demonstrou na realização desta dissertação.





# Resumo

---

Desde sempre houve uma preocupação das pessoas com a segurança, quer com a sua vida, quer com a dos seus bens. Esta preocupação, levou a que ao longo do tempo fossem desenvolvidos sistemas de vigilância que ajudam a alertar para eventuais perigos. Estes sistemas têm vindo a evoluir ao longo do tempo, no entanto, ainda existem aspetos em que podem ser melhorados, tais como: a Cooperação entre humanos e os dispositivos de vigilância; a Descentralização dos Sistemas; a Fusão da informação sensorial; ou a Reconfiguração e Dinamismo dos sistemas.

Tendo em conta estas lacunas, neste trabalho é proposta uma arquitetura de Sistemas Inteligentes de Vigilância baseada em agentes de Software georreferenciados, com vista a aumentar a cooperação entre humanos e sensores. Esta arquitetura pretende melhorar o desempenho global dos sistemas de vigilância atuais utilizando a localização dos recursos humanos e a localização dos eventos, para melhorar os tempos de resposta a eventos.

Com base na arquitetura proposta, foi desenvolvido um sistema que permitiu testar a solução proposta. Os resultados obtidos através dos vários testes efetuados, permitiram comprovar que, entre outras melhorias, a arquitetura proposta permite uma redução nos tempos de resposta aos eventos.

**Palavras-chave:** Multiagentes, sensores, Geolocalização, Sistemas de Vigilância, Arquitetura, Dispositivos Móveis, Cooperação.

---



# Abstract

---

There has always been a concern of people with security, both with their life and with their property. This concern has led to the development of surveillance systems over time that help to alert to possible dangers. These systems have been evolving over time, however, there are still areas where they can be improved, such as: Human-to-human cooperation with surveillance devices; Decentralization of the Systems; Fusion of sensory information; or Reconfiguration and Dynamism of systems.

Considering these gaps, this work proposes an architecture of Intelligent Surveillance Systems based on georeferenced software agents, to increase the cooperation between humans and sensors. This architecture aims to improve the overall performance of current surveillance systems using the location of human agents and the location of events to improve response times to events.

Based on the proposed architecture, a system was developed that allowed to test the proposed solution. The results obtained through the various tests carried out showed that, among other improvements, the proposed architecture allows a reduction in the response times to the events.

**Keywords:** Multi-agents; sensors; surveillance system; geographic position; architecture; mobile devices; cooperation

---



# Conteúdo

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1 QUESTÃO DE INVESTIGAÇÃO .....	5
1.2 HIPÓTESE E ABORDAGEM.....	6
<b>2. REVISÃO DO ESTADO DA ARTE.....</b>	<b>9</b>
2.1 CONCEITOS GERAIS.....	16
2.1.1 <i>Agentes</i> .....	18
2.2 CONSTITUIÇÃO DOS SISTEMAS DE VIGILÂNCIA.....	22
2.3 ARQUITETURAS DE SISTEMAS DE VIGILÂNCIA.....	24
2.3.1 <i>Arquitetura centralizada</i> .....	24
2.3.2 <i>Arquitetura distribuída</i> .....	28
2.4 ANÁLISE COMPARATIVA .....	36
<b>3. METODOLOGIA DE INVESTIGAÇÃO.....</b>	<b>39</b>
3.1 CONTRIBUIÇÃO ESPERADA.....	40
3.2 PLANO DE TRABALHO .....	42
3.3 METODOLOGIA DE VALIDAÇÃO.....	45
3.4 PRINCIPAIS CONTRIBUIÇÕES E PUBLICAÇÕES.....	49
3.5 INTEGRAÇÃO COM OUTRAS ATIVIDADES DE INVESTIGAÇÃO .....	51
<b>4. ARQUITETURA BASEADA EM MULTIAGENTES</b>	
<b>GEORREFERENCIADOS PARA SISTEMAS DE VIGILÂNCIA.....</b>	<b>53</b>
4.1 ARQUITETURA.....	54
4.2 COMPONENTES DA ARQUITETURA .....	56

4.3	MODELO DE DADOS.....	63
4.3.1	<i>Parâmetros.....</i>	63
4.3.2	<i>Regras.....</i>	63
4.3.3	<i>Eventos.....</i>	64
4.3.4	<i>Ações.....</i>	67
4.4	FUNCIONAMENTO DO SISTEMA.....	68
4.4.1	<i>Atribuição georreferenciada dinâmica de um Recurso a um Evento.....</i>	75
4.4.2	<i>Processo de Seguimento de Intrusos.....</i>	84
5.	<b>IMPLEMENTAÇÃO DA ARQUITETURA.....</b>	<b>87</b>
5.1	PLATAFORMA DE AGENTES .....	87
5.2	INTEGRAÇÃO DE SENSORES.....	89
5.3	SISTEMA DE AGENTES.....	91
5.4	PROTÓTIPO DESENVOLVIDO .....	95
5.4.1	<i>Interface web.....</i>	96
5.4.2	<i>Aplicação mobile.....</i>	100
5.4.3	<i>Sensores.....</i>	101
5.4.4	<i>Processamento .....</i>	103
6.	<b>TESTES E RESULTADOS .....</b>	<b>105</b>
6.1	CENÁRIO.....	105
6.2	SIMULAÇÕES .....	110
6.2.1	<i>Caso 1 – 10 Recursos, 10 Eventos.....</i>	113
6.2.2	<i>Caso 2 – 5 Recursos, 10 Eventos.....</i>	118
6.3	TESTES EM AMBIENTE REAL.....	120
6.3.1	<i>Resultados.....</i>	122
7.	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>125</b>
7.1	TRABALHO FUTURO .....	128
8.	<b>REFERÊNCIAS .....</b>	<b>131</b>

## Lista de Figuras

FIGURA 1.1 - COMO AUMENTAR A COOPERAÇÃO ENTRE AGENTES? .....	6
FIGURA 2.1 – EXEMPLO DE APLICAÇÕES POR TIPO DE TECNOLOGIA - ADAPTADO DE [21]. .....	14
FIGURA 2.2 – SISTEMA FEDERADO [39] .....	19
FIGURA 2.3 –FIPA <i>COMMUNICATION ACTS</i> [40].....	20
FIGURA 2.4 – FIPA <i>INTERACTION PROTOCOLS</i> [40].....	21
FIGURA 2.5 – BLOCOS QUE COMPÕEM TRADICIONALMENTE OS SISTEMAS DE VIGILÂNCIA USANDO A VISÃO .....	22
FIGURA 2.6 – ARQUITETURA DO SISTEMA PRISMATICA [19] .....	25
FIGURA 2.7 – ARQUITETURA DO SISTEMA ADVISOR [59].....	26
FIGURA 2.8 – ARQUITETURA DO SISTEMA AMFIS [60].....	27
FIGURA 2.9 – ARQUITETURA DO SISTEMA APRESENTADO EM [61] .....	28
FIGURA 2.10 – ARQUITETURA BASEADA EM AGENTES PARA MONITORIZAÇÃO DESCRITA EM [4] .....	29
FIGURA 2.11 – ARQUITETURA MULTIAGENTES [63].....	30
FIGURA 2.12 – ARQUITETURA DO <i>HARDWARE</i> . [47].....	32
FIGURA 2.13 – ARQUITETURA DO SOFTWARE DAS CÂMARAS INTELIGENTES [68] .....	33
FIGURA 2.14 – ARQUITETURA DO SISTEMA APRESENTADO EM [70].....	34
FIGURA 2.15 – ARQUITETURA DO SISTEMA APRESENTADO EM [71].....	35
FIGURA 2.16 – ARQUITETURA DO SISTEMA OCULUS [74] .....	36
FIGURA 3.1 – METODOLOGIA DE INVESTIGAÇÃO ADAPTADO DE [75].....	39
FIGURA 3.2 – PLANO DE TRABALHO .....	42
FIGURA 4.1 – ARQUITETURA DO SISTEMA .....	55
FIGURA 4.2 - CADEIA DE PROCESSAMENTO DE INFORMAÇÃO.....	56
FIGURA 4.3 – FLUXO DE INFORMAÇÃO DO AGENTE SENSOR.....	57
FIGURA 4.4 – FLUXO DE INFORMAÇÃO DO AGENTE PROCESSADOR.....	58

FIGURA 4.5 – FLUXO DE INFORMAÇÃO DO AGENTE INFERÊNCIA.....	59
FIGURA 4.6 – FLUXO DE INFORMAÇÃO DO AGENTE AÇÃO.....	59
FIGURA 4.7 – FLUXO DE INFORMAÇÃO DO AGENTE MOBILE.....	60
FIGURA 4.8 – FLUXO DE INFORMAÇÃO DO AGENTE INTERFACE.....	61
FIGURA 4.9 – FLUXO DE INFORMAÇÃO DO AGENTE BACKUP.....	62
FIGURA 4.10 – FLUXO DE INFORMAÇÃO DO AGENTE MONITORIZAÇÃO.....	62
FIGURA 4.11 – PARÂMETROS.....	63
FIGURA 4.12 – EVENTOS.....	65
FIGURA 4.13 – DIAGRAMA DE ESTADOS DOS EVENTOS.....	66
FIGURA 4.14 – GERAÇÃO DE PARÂMETROS.....	69
FIGURA 4.15 – GERAÇÃO DE AÇÕES.....	70
FIGURA 4.16 – ARQUITETURA ESTENDIDA [83].....	74
FIGURA 4.17 – DIAGRAMA DE ESTADOS DE ATRIBUIÇÃO DE EVENTOS.....	78
FIGURA 4.18 – RESUMO DO PROCESSO DE SEGUIMENTO DE SUSPEITOS.....	85
FIGURA 5.1 – COMUNICAÇÃO AGENTES - SENSORES.....	90
FIGURA 5.2 – DIAGRAMA DE SENSORES.....	90
FIGURA 5.3 – MÓDULO MOBILE.....	91
FIGURA 5.4 – MÓDULO INTERFACE.....	92
FIGURA 5.5 – EDITOR DE REGRAS.....	95
FIGURA 5.6 – EXEMPLO DE VISUALIZAÇÃO DO DETALHE DE UM SENSOR.....	96
FIGURA 5.7 – EXEMPLO DA CRIAÇÃO DE UM EVENTO.....	97
FIGURA 5.8 – EXEMPLO DETALHE DE UM EVENTO.....	97
FIGURA 5.9 – EXEMPLO MONITORIZAÇÃO DE CÂMARAS.....	98
FIGURA 5.10 – EXEMPLO DE VISUALIZAÇÃO DE PARÂMETROS E EVENTOS.....	99
FIGURA 5.11 – MÓDULO MOBILE – ECRÃ INICIAL.....	100
FIGURA 5.12 – MÓDULO MOBILE – CONSULTA DO MAPA.....	100
FIGURA 5.13 – MÓDULO MOBILE – CRIAÇÃO DE UM EVENTO.....	101
FIGURA 5.14 – MÓDULO MOBILE – EVENTOS ATRIBUÍDOS.....	101
FIGURA 5.15 – EXEMPLO 1 DE MONTAGEM DVAs.....	102
FIGURA 5.16 - EXEMPLO 2 DE MONTAGEM DE DVAs.....	103
FIGURA 6.1 – IDENTIFICAÇÃO DOS LOCAIS.....	106
FIGURA 6.2 – MAPA COM A LOCALIZAÇÃO DOS EVENTOS E RECURSOS.....	112
FIGURA 6.3 – EXEMPLO DO MAPA DOS EVENTOS.....	120
FIGURA 6.4 – EXEMPLO DE DETECÇÃO DE INCÊNDIO [93].....	123
FIGURA 6.5 – EXEMPLO DE DETECÇÃO DE INTRUSOS E SEGUIMENTO.....	123



## Lista de Tabelas

TABELA 2.1 - CARACTERÍSTICAS DA PRIMEIRA GERAÇÃO DOS SISTEMAS DE VIGILÂNCIA [1] .....	11
TABELA 2.2 - CARACTERÍSTICAS DA SEGUNDA GERAÇÃO DOS SISTEMAS DE VIGILÂNCIA [1] .....	12
TABELA 2.3 - CARACTERÍSTICAS DA TERCEIRA GERAÇÃO DOS SISTEMAS DE VIGILÂNCIA [1] .....	13
TABELA 3.1 - CARACTERÍSTICAS A ANALISAR .....	48
TABELA 4.1 - MENSAGENS ENTRE AGENTES .....	73
TABELA 4.2 – PERFIL VERSOS PERFORMANCE .....	82
TABELA 6.1 – DISTRIBUIÇÃO DOS SENSORES POR LOCAL .....	106
TABELA 6.2 – EXEMPLO DE PARÂMETROS DEFINIDOS .....	108
TABELA 6.3 – EXEMPLO DE REGRAS DEFINIDAS .....	109
TABELA 6.4 – RECURSOS USADOS NAS SIMULAÇÕES .....	111
TABELA 6.5 – EVENTOS USADOS NAS SIMULAÇÕES .....	111
TABELA 6.6 – RESULTADOS DAS SIMULAÇÕES COM 10 RECURSOS .....	115
TABELA 6.7 – RESULTADOS DAS SIMULAÇÕES COM 10 RECURSOS (CONTINUAÇÃO) .....	116
TABELA 6.8 – COMPARAÇÃO DE TEMPOS DE CHEGADA AOS EVENTOS POR SIMULAÇÃO COM 10 RECURSOS .....	117
TABELA 6.9 – COMPARAÇÃO DE TEMPOS DE CHEGADA AOS EVENTOS POR SIMULAÇÃO COM 5 RECURSOS .....	118
TABELA 6.10 – RESULTADOS DAS SIMULAÇÕES COM 5 RECURSO .....	119
TABELA 6.11 – LISTA DE EVENTOS .....	121
TABELA 6.12 – SITUAÇÃO ATUAL DOS RECURSOS .....	121
TABELA 6.13 – CÁLCULO PARA A CLASSIFICAÇÃO DOS RECURSOS .....	122



## Lista de Acrónimos

2D – Duas dimensões

3D – Três dimensões

ACL - Agent Communication Language

AIPS - Agent Interaction Protocol Suite

AMS - Agent Management System

CAs - Communicative Acts

C-MSF – Centralized Multi-sensor fusion

CCD – Charge Coupled Device

CCTV – Closed-Circuit Television

DF - Directory Facilitator

DTW – Dynamic Time Warping

FIPA - Foundation for Intelligent Physical Agent

MSF – Multi-sensor fusion

CMOS – Complementary metal-oxide semiconductor

D-MSF – Distributed Multi-sensor fusion

DVA – Sistema Avançado de Vigilância baseado em agentes - sistema Inteligente de Vigilância recorre a um conjunto de componentes autónomos (sensores, câmaras, Tablets/Telemóveis) que cooperam entre si e com recursos humanos num modelo integrado de vigilância capaz de exibir Inteligência Coletiva no desempenho de funções de vigilância e segurança.

GPS – Global Positioning System

HMM – Hidden Markov Models

JADE – JAVA Agent DEvelopment Framework

JADE-LEAP – Lightweight Extensible Agent Platform

JPEG – Joint Photographic Experts Group

LiDAR – Light Detection And Ranging

PTZ – Pan–Tilt–Zoom

QREN – Quadro de Referência Estratégica Nacional

SERVROBOT – Robô de serviço (*Service Robot*) reconfigurável, todo o terreno, com comportamentos autónomos é um robô adaptável a diferentes tipos de utilização e condições ambientais, capaz de recolher informação sensorial de forma local ou remota e de atuar autonomamente com base nessa informação.

SI I&DT – Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico

SQL – Structured Query Language

SSE – *Plug-and-play* de câmaras inteligentes

XML – Extensible Markup Language

# 1

## Introdução

Desde sempre, a preocupação com a segurança de pessoas e bens, tem vindo a fazer parte das sociedades. Paralelamente a esta preocupação, e juntamente com a evolução tecnológica, tem existido uma evolução na vigilância, passando-se da vigilância feita apenas por pessoas, para o uso de Sistemas de Vigilância que integram vários algoritmos capazes de aumentar o nível de segurança dos locais sob vigia e reduzindo a dependência dos humanos, passando pelos, ainda, tradicionais sistemas CCTV (*Closed-Circuit Television*) [1].

Inteligência pode definir-se como a capacidade para aprender, perceber e se adaptar a um determinado ambiente usando o seu próprio conhecimento [2]. Dotando os Sistemas de Vigilância de Inteligência, podem-se evitar consequências gravosas sobre pessoas ou bens, decorrentes de ações intencionais de vandalismo, ou furto, ou de eventos disruptivos como incêndios, fugas de gases, terremotos ou inundações. Através da deteção automática, é possível identificar ocorrências anormais de eventos monitorizados por um sistema de segurança, permitindo que alertas sejam ativados apenas quando existem suspeitas fundamentadas (e eventualmente confirmadas por operadores) da ocorrência de eventos não desejados.

Embora existam diversos desenvolvimentos e progresso nos atuais Sistemas de Vigilância, existe ainda muito a melhorar nestes sistemas [3]. Algumas

das características em que estes sistemas podem ser melhorados são identificados por Xiaogang Wang em [3], M. Valera e S.A. Velastin em [1] e por Javier Albusac et al. em [4]. Tendo em conta as lacunas identificadas, alguns dos aspetos que podem ser melhorados estão relacionados com a diminuição da dependência destes sistemas dos humanos, a melhoria dos algoritmos para identificação de eventos, a redução da dependência de nós centrais de processamento, a diversificação do tipo de sensores empregues (ou limitações do número de sensores ligados ao sistema) e uma melhor interação humano-máquina.

Em relação à dependência dos Sistemas de Vigilância da intervenção humana, tem existido um esforço para sua redução com a introdução de algoritmos que permitem a identificação de eventos, sem a necessidade de os humanos estarem constantemente a olhar para monitores ou a fazer rondas pelos locais a vigiar. No entanto, ainda existem pontos onde esta dependência pode ser reduzida e outros pontos onde a cooperação entre os humanos e os sistemas pode ser melhorada; como por exemplo a utilização da localização atual dos agentes de segurança em relação à localização do evento para uma resposta mais eficaz ao mesmo.

Embora em alguns casos exista a hipótese de integrar diferentes componentes, a integração é efetuada na fase de conceção de sistema, tornando-se a sua reconfiguração difícil. As soluções existentes passam também pelo desenho de soluções "chave-na-mão" e pelo desenvolvimento, nalguns casos pontuais, de soluções específicas de integração das várias funcionalidades.

Refira-se, a título de exemplo, que nas soluções atuais são usualmente propostos sensores independentes para a deteção de gases tóxicos (e.g. dióxido de carbono) e de presença de pessoas nas instalações; porém, a junção destes dois tipos de informação é crítica para decidir o curso de ações a realizar, pois um incremento do nível de dióxido de carbono pode não ser crítico se não existirem pessoas nas instalações.

No que se refere às arquiteturas usadas como base destes sistemas continuam, na sua grande maioria, a ser baseadas em soluções centralizadas e de di-

fácil reconfiguração [5]. Existiu nos últimos anos um esforço da comunidade científica em desenvolver novas arquiteturas que permitam produzir sistemas distribuídos.

Algumas das arquiteturas distribuídas propostas, exemplificadas no capítulo 2, baseiam-se em soluções modulares, onde o processamento é repartido por várias unidades de processamento, outras adotam os agentes de software como meio de abordagem ao problema. Tem existido uma contínua inovação nesta área, sendo que, alguns dos pontos em que comunidade científica se tem focado são[1]:

- Adaptação, fusão de dados e métodos de seguimento em ambientes cooperativos com multisensores;
- Extensão das técnicas para identificação e classificação de objetos;
- Novos protocolos de comunicação e modelos de metadados na comunicação e integração entre diferentes módulos;
- Desenvolvimento de arquiteturas para sistemas multissensoriais, com algoritmos de processamento em tempo real e capacidade de reconfiguração;
- Arquiteturas adaptáveis a diferentes ambientes.

Por outro lado, a proliferação de dispositivos móveis com acesso à internet e com sensores GPS (*Global Positioning System*) integrados tem acompanhado o crescimento das Redes de comunicação de Nova Geração, permitindo o acesso remoto a informação e a identificação do local onde a pessoa se encontra. Estes avanços, aliados à melhoria de desempenho dos computadores e à sua miniaturização, permitem definir novas abordagens aos sistemas de vigilância.

Desta forma, o trabalho desenvolvido no âmbito do doutoramento, visa contribuir com o desenvolvimento de uma arquitetura para um Sistema de Vigilância baseado em Agentes Georreferenciados. O trabalho proposto pretende

tirar partido das vantagens tecnológicas introduzidas pelas Redes de comunicação de Nova Geração, como as elevadas taxas de transferência de dados, e pela proliferação de dispositivos GPS de baixo custo, permitindo incorporar na arquitetura o conceito de componentes autónomos (sensores, telemóveis) que, dotados de capacidade de processamento local e comunicação, poderão cooperar entre si para a criação de um sistema integrado de vigilância capaz de exibir Inteligência Coletiva<sup>1</sup> no desempenho de funções de vigilância e segurança.

A arquitetura proposta permitirá, por exemplo, melhorar a vigilância de uma grande área (ex.: um perímetro fabril) onde a deteção de eventos seja autónoma e adaptável. Isto é, o sistema pode facilmente passar de uma vigilância estática, onde apenas procura potenciais eventos, para uma vigilância dinâmica, onde adapta o seu comportamento de acordo com o evento detetado. Sendo assim, se determinado recurso identifica um evento específico os restantes reorganizar-se-ão de forma a colmatar a sua falta no sistema global, para que não haja um decréscimo de desempenho.

Outro exemplo de utilização deste tipo de abordagem é a deteção de intrusos, em que os agentes de forma individual ou coletiva (através de sensores, por processamento de imagem, ou da cooperação entre vários tipos de sensores) detetam o intruso e interagem entre si (tirando partido de algoritmos baseados em localização). Assim sendo, o sistema permite transmitir informações ao(s) agente(s) mais adequados para realizarem o seu processamento, ao mesmo tempo que segue automaticamente o intruso, cooperando, por exemplo, com outras câmaras que se encontrem na direção em que este se desloca, quando sai do campo de visão da câmara atual.

---

<sup>1</sup> Inteligência Coletiva - Grupos de entidades que fazem coisas coletivamente que parecem inteligentes [2]



## 1.1 *Questão de investigação*

É com base nas limitações descritas acima e com uma análise do atual estado da arte, descrita no capítulo seguinte, que foi levantada a questão de investigação:

**“Tendo em conta as novas tecnologias disponíveis atualmente,  
como se podem melhorar os atuais sistemas de vigilância?”**

A questão de investigação levantada focou-se na necessidade sentida em melhorar os atuais Sistemas de Vigilância, questionando se estes poderiam ser melhorados através da implementação de novas abordagens na comunicação e colaboração entre agentes (humanos e sensores); procurando aumentar o nível de sucesso, quer na identificação de eventos, quer na reação aos eventos, pela otimização da utilização dos recursos disponíveis e pela partilha de informação e contexto.

Muitos dos atuais Sistemas de Vigilância ainda são baseados na monitorização de locais através da visualização de vídeo, outros sistemas mais avançados já incorporam algoritmos para a deteção automática de eventos lançando um alerta para um sistema central que notifica os responsáveis pela segurança da deteção de um evento. Embora estes sistemas tenham evoluído bastante, ainda deixam de lado alguns aspetos que podem melhorar a eficiência dos sistemas de vigilância, como por exemplo, a utilização da localização do evento e dos recursos responsáveis pela vigilância, o tipo de evento, a carga atual dos recursos disponíveis ou a interação durante a resolução do evento.

Neste sentido, a linguagem e a cooperação entre agentes em sistemas de vigilância, permitem apresentar um novo paradigma nesta área, centrado na comunicação e integração de agentes em sistemas distribuídos. Recorrendo esta melhoria na comunicação e integração, podem desenvolver-se sistemas *Plug-and-Play*, onde seja possível incluir um novo agente no sistema sem a necessidade de reconfiguração.



Figura 1.1 - Como aumentar a cooperação entre agentes?

Resumindo, tal como ilustrado na Figura 1.1, existem vários componentes nos Sistemas de Vigilância, tais como os humanos, os sensores, os algoritmos de seguimento, os algoritmos de deteção de objetos e eventos e os alertas. Foi necessário definir uma arquitetura que permita tirar o máximo partido destes componentes, surgindo assim a questão de investigação enunciada.

## 1.2 *Hipótese e abordagem*

Tendo como base a questão de investigação, foi formulada a seguinte hipótese:

**“Uma arquitetura de Sistemas Inteligentes de Vigilância, baseada em agentes de Software georreferenciados, pode aumentar a cooperação entre humanos e sensores, melhorando assim o desempenho global dos sistemas de vigilância atuais.”**

Através do estudo do estado da arte e com a identificação de requisitos por parte de empresas de segurança, abordou-se o problema descrito na secção anterior, desenhando uma arquitetura baseada em agentes de software georre-

ferenciados que permite melhorar a cooperação entre humanos e sensores. Esta arquitetura descentralizada, tira partido das novas tecnologias (por exemplo o GPS dos dispositivos móveis) para obter a localização geográfica dos seus recursos. Com os recursos e eventos georreferenciados são aplicados mecanismos para atribuição dinâmica dos eventos que permitem melhorar a resposta a eventos através da cooperação entre sistema e humanos.

Neste trabalho, estão interligadas várias áreas de saber (Monitorização, Georreferenciação, Distribuição automática de tarefas e Vigilância Distribuída), que não se encontram normalmente integradas. De facto, na literatura é possível encontrar trabalhos nas áreas de sistemas distribuídos de videovigilância, trabalhos na área de redes de sensores, trabalhos na área de sistemas de monitorização e trabalhos na área da distribuição automática de tarefas. Porém, uma solução distribuída, envolvendo agentes inteligentes, máquinas e humanos que cooperam entre si, como a presente proposta, integrando valências de todas estas áreas não existe atualmente na vigilância. [6]

A validação da hipótese colocada foi efetuada pela construção de um Sistema de Vigilância Inteligente baseado na arquitetura definida, onde foi possível avaliar as suas vantagens contra as atuais arquiteturas.

A utilização de um sistema baseado nesta arquitetura e aplicado a um cenário real, permitiu verificar as vantagens da cooperação em tempo real entre recursos, permitindo aumentar o desempenho global.



## Revisão do estado da Arte

O instinto de sobrevivência está presente em todos os seres vivos, tendo cada um deles criado mecanismos para a assegurar. No caso dos humanos, esta preocupação está muito mais presente, uma vez que para além da sua maior capacidade para consciencialização do perigo, existe também uma grande preocupação com a segurança dos seus bens.

Esta preocupação com a segurança de pessoas e bens faz parte da sociedade desde sempre, tendo vindo a ser cada vez mais importante e orientando-se para a utilização de meios cada vez mais sofisticados para a assegurar. Os primeiros humanos tinham preocupações muito básicas, sendo que a sua sobrevivência era a sua principal preocupação. Numa segunda fase, quando passou de caçador/coletor para a agricultura, criação gado e a ter casas, o Homem passou a preocupar-se com segurança destes também. Ao longo da sua evolução, o Homem, à medida que foi aumentando e diversificando os seus bens foi tendo cada vez mais a necessidade de os segurar.

Como resposta a esta necessidade, o Homem começou a vigiar e, embora estas práticas de vigilância existam desde os primórdios da sociedade, é na sociedade da informação que a vigilância, tal como a conhecemos, emergiu como uma componente central da vida moderna [7]. Existem várias definições para vigilância, como por exemplo, o *Concise Oxford Dictionary* define vigilância co-

mo “observação atenta, especialmente de uma pessoa suspeita” [8], William Staples em *“The Surveillance Studies Reader”* como “a ação de manter um olhar atento sobre pessoas” [9] e David Lyon em *“Surveillance studies: An overview”* como a “atenção focada, sistemática e rotineira de dados pessoais para fins de influência, gestão, proteção ou direção” [7]. Vigilância também pode ser definida como a recolha, análise e interpretação de dados de forma sistemática, permitindo a identificação de eventos [10][11].

A vigilância é focada e é sistemática, isto é, a atenção aos dados não é aleatória, ocasional ou espontânea; é deliberada e depende de certos protocolos e técnicas. A vigilância também é caracterizada pela rotina. Ocorre como parte da vida quotidiana em todas as sociedades que dependem de administração burocrática e de alguns tipos da tecnologia da informação. Vigilância quotidiana é endêmica para as sociedades modernas. [7]

Para melhorar/facilitar a vigilância surgiu a necessidade de criar sistemas de vigilância. Os primeiros esforços relacionados com a produção de sistemas de videovigilância, que são os precursores dos que hoje existem, foram feitos em 1942 na Alemanha, quando a Siemens AG em Peenemunde instalou pela primeira vez um *“Closed Circuit Television”* (CCTV) – Circuito fechado de televisão para monitorização de testes feitos com mísseis V-2. No entanto, só mais tarde em 1949, foi comercializado o primeiro CCTV, chamado Vericon [12]. O uso destes sistemas foi difundido a partir dos anos 70, quando começou a ser possível a gravação em tempo real em cassetes, facilitando assim o acesso à informação histórica [13].

A evolução dos Sistemas de Vigilância pode ser caracterizada em três gerações [14]. A primeira geração (1949-1980) era categorizada por sistemas analógicos, que permitiam a monitorização de locais usando várias câmaras ligadas a diferentes monitores ou várias câmaras ligadas a um só monitor, usando um comutador que permitia alternar entre câmaras. Estes sistemas dependiam de humanos para a deteção de possíveis eventos, sendo que estes só eram detetados caso fossem “observados” no monitor pelo responsável de segurança.

Estes sistemas usavam normalmente câmaras com sensores digitais CCD (*Charge Coupled Device*) para captura da imagem que, depois de convertida para analógico, era enviada, por cabos coaxiais, para um monitor. Na Tabela 2.1 é feito um resumo desta primeira geração em termos de técnicas usadas, vantagens, problemas e o estado atual da investigação, baseada nesta tecnologia.

	Primeira Geração
Técnicas	<ul style="list-style-type: none"> <li>Sistemas analógicos de CCTV.</li> </ul>
Vantagens	<ul style="list-style-type: none"> <li>Bom desempenho;</li> <li>Tecnologia robusta.</li> </ul>
Problemas	<ul style="list-style-type: none"> <li>Comunicação e armazenamento de imagens em formato analógico.</li> </ul>
Estado atual de investigação	<ul style="list-style-type: none"> <li>Digital versus Analógico;</li> <li>Gravação digital de vídeo;</li> <li>Compressão de vídeo CCTV.</li> </ul>

**Tabela 2.1 - Características da Primeira Geração dos Sistemas de Vigilância [1]**

A excessiva dependência dos humanos nos Sistemas de Vigilância e a necessidade de melhorar a detecção de eventos levou ao aparecimento de novas soluções. O surgimento da digitalização de imagens permitiu a introdução de algoritmos que permitem a análise das imagens recolhidas e a detecção de possíveis situações de risco. Este avanço nos Sistemas de Vigilância é considerado como a Segunda Geração (1980-2000) destes sistemas [14].

	Segunda geração
Técnicas	<ul style="list-style-type: none"> <li>Sistemas de Vigilância Automáticos com recurso a algoritmos robustos combinados com sistemas de CCTV.</li> </ul>
Vantagens	<ul style="list-style-type: none"> <li>Aumento da eficiência dos sistemas de CCTV.</li> </ul>
Problemas	<ul style="list-style-type: none"> <li>Necessidade de algoritmos robustos de detecção e seguimento para análise de comportamentos.</li> </ul>

Estado atual de investigação	<ul style="list-style-type: none"> <li>• Algoritmos de visão computacional em tempo real robustos;</li> <li>• Aprendizagem automática de cenários e comportamentos;</li> <li>• Interligação entre a análise estatística de cenários, a produção e sua interpretação.</li> </ul>
------------------------------	---

**Tabela 2.2 - Características da Segunda Geração dos Sistemas de Vigilância [1]**

Com a introdução de algoritmos passou a ser possível a detecção automática de pessoas e objetos, permitindo assim analisar comportamentos. Na Tabela 2.2 é apresentado um resumo do que caracteriza esta segunda geração.

Por último, a Terceira Geração caracteriza-se pelo incremento na área de monitorização dos Sistemas de Vigilância, através da introdução de novas tecnologias, inserção de outros tipos de sensores e a distribuição da informação entre os vários elementos do sistema. Com o aumento das áreas a vigiar e a maior abrangência de situações a detetar, o mercado sentiu a necessidade de melhorar os sistemas existentes, quer a nível de processos, quer introduzindo mais informação no sistema.

A introdução de novos sensores, como forma de complemento às câmaras CCTV, levou ao desenvolvimento de técnicas de fusão de dados permitindo correlacionar informação. No entanto, esta inclusão aumentou a complexidade de integração, quer a nível de hardware, quer de software.

Por outro lado, os sistemas centralizados, caracterizados por um ponto central (normalmente um único computador) apresentam algumas limitações, o que levou também à abordagem de sistemas que permitam a descentralização, distribuindo assim a carga de processamento e a dependência de um só ponto de falha. Na Tabela 2.3, é feita a caracterização da terceira geração.



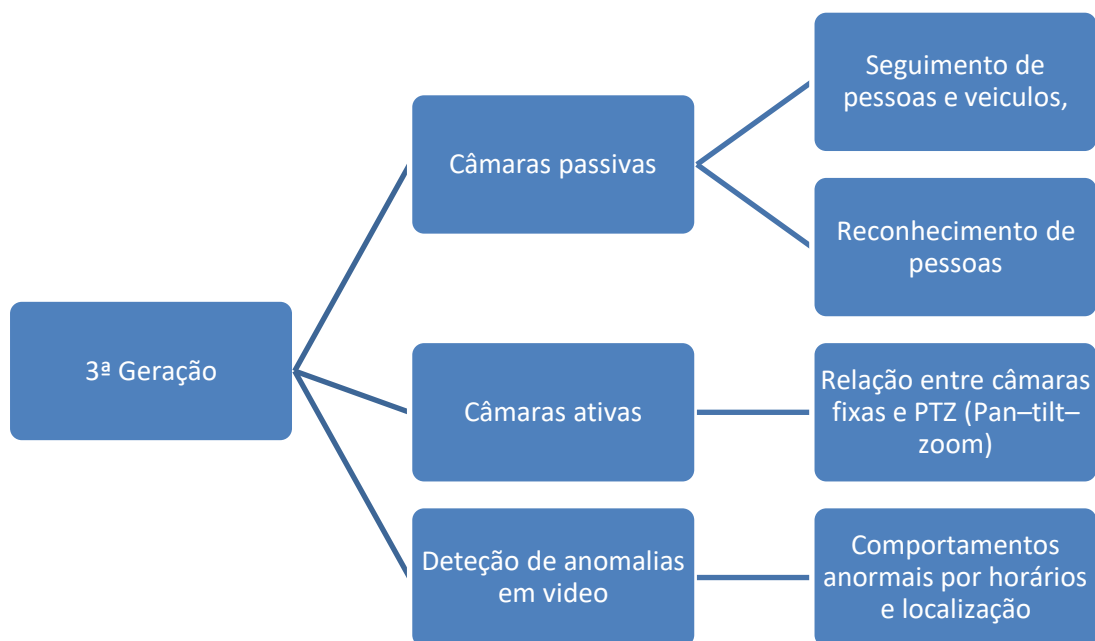
	Terceira Geração
Técnicas	<ul style="list-style-type: none"> <li>• Sistemas de Vigilância Automáticos<sup>2</sup> para grandes áreas.</li> </ul>
Vantagens	<ul style="list-style-type: none"> <li>• Aumento da informação resultante da combinação de vários sensores;</li> <li>• Distribuição.</li> </ul>
Problemas	<ul style="list-style-type: none"> <li>• Partilha da informação (integração e comunicação);</li> <li>• Metodologias de design;</li> <li>• Plataformas móveis e plataformas multisensor.</li> </ul>
Estado atual de investigação	<ul style="list-style-type: none"> <li>• Inteligência central <i>versus</i> distribuída;</li> <li>• Plataformas de critério probabilístico;</li> <li>• Técnicas de vigilância multi-câmara.</li> </ul>

**Tabela 2.3 - Características da Terceira Geração dos Sistemas de Vigilância [1]**

As aplicações desta Terceira Geração são diversas, sendo que cada tipo de tecnologia está mais direcionada para uma utilização específica, como é exemplificado na Figura 2.1. Com esta diversidade de aplicações, a terceira geração foi utilizada, desde o seu aparecimento, numa grande variedade de cenários, como mostram alguns exemplos que surgiram logo com o seu aparecimento: Aeroportos [15][16]; Portos [17]; Metros [18][19] e Tráfego [20].

---

<sup>2</sup> Sistemas que combinam novas técnicas de processamento de imagem e comunicação de vídeo digital com sistemas CCTV [21]



**Figura 2.1 – Exemplo de aplicações por tipo de tecnologia** - Adaptado de [21].

Nos últimos anos com o aparecimento de novas tecnologias, como os drones, estão a ser desenvolvidas e incorporadas novas abordagens, permitindo a disponibilização de novas funcionalidades e o aumento da eficácia destes sistemas [22]. Esta evolução, sentida ao longo dos anos, nestes sistemas é apoiada sobretudo pela sua grande proliferação no mercado, abrangendo uma vasta gama de áreas, como a vigilância de transportes públicos, de superfícies comerciais, de áreas industriais, de casas inteligentes, etc.

Um dos exemplos do uso de novas tecnologias em sistemas de monitorização, está a ser desenvolvido no âmbito do projeto RIVERWATCH<sup>3</sup>, onde o uso de vários robôs autónomos num sistema de monitorização ambiental permite a vigilância de cursos de água.

Também na área da saúde encontramos sistemas de vigilância, que embora não sejam caracterizados pelos parâmetros mais comuns (câmaras e outros tipos de sensores), não deixam de ser mais um exemplo da sua importância pa-

---

<sup>3</sup> ECHORD RIVERWATCH - Cooperating robots for monitoring of riverine environments - <http://www.echord.info/wikis/website/riverwatch>

ra a sociedade de hoje em dia. Dois exemplos desta difusão pela área da saúde são: "*Surveillance and response: Tools and approaches for the elimination stage of neglected tropical diseases*" [23], onde é apresentado um estudo de sistemas de vigilância aplicados na área de saúde, nomeadamente nas doenças negligenciadas tropicais; e em "*Surveillance systems evaluation a systematic review of the existing approaches*" [24], onde é feita uma revisão dos métodos de avaliação dos sistemas de vigilância na área da saúde.

## 2.1 *Conceitos gerais*

Nesta seção é feita uma breve descrição de alguns conceitos usados ao longo desta dissertação.

Um destes conceitos é Sistema, que provém do latim *Systema* e que significa “todo complexo; todo composto por partes; harmonia” [25]. Centrando a atenção no Sistema de Software, pode ser definido como um conjunto de componentes de software que interagem ou interdependentes e que formam um todo integrado [26].

Os Sistemas de Software cujos componentes são executados em apenas um computador e que não comuniquem com outros computadores são **Sistemas Centralizados**. Se componentes idênticos ou semelhantes forem executados em diferentes computadores trata-se de um **Sistema Descentralizado**. A principal diferença entre um sistema centralizado e um descentralizado é a forma como os componentes comunicam e se sincronizam, sendo que no centralizado é feito através da partilha de memória do computador e no descentralizado é feita através do envio de mensagens. [27]

Um **Sistema Distribuído** é um conjunto de computadores independentes que comunicam entre si e que aparecem ao utilizador do sistema como um único computador [28]. Os elementos de software ou hardware estão integrados numa rede de computadores que comunicam e se coordenam através do envio de mensagens [29].

Existem inúmeras diferenças entre os sistemas centralizados e os distribuídos, começando pelo acesso aos dados, em que os sistemas centralizados têm uma maior facilidade de acesso, uma vez que estes estão normalmente localizados no mesmo local. Por outro lado, os sistemas distribuídos permitem abarcar áreas geográficas de maior dimensão, dada a possibilidade de acesso remoto. Em relação às tecnologias e processos, também existem algumas diferenças, uma vez que os sistemas centralizados são caracterizados por tecnologias e processos comuns que facilitam a gestão e consistência destes sistemas. Os sistemas distribuídos podem interligar diferentes tecnologias e distribuir processos por

diferentes localizações, permitindo a modularidade, escalabilidade do sistema e a partilha de informação. Os sistemas distribuídos também permitem uma degradação graciosa[30], aumentando assim a sua confiabilidade e disponibilidade, uma vez que as falhas totais do sistema são reduzidas. [31]

**Arquitetura de Software** descreve a organização e interação entre os componentes de software, focando-se na organização lógica do software [28].

**Arquitetura de Sistema** descreve a localização dos componentes de software nos computadores físicos. Quando os componentes estão localizados num único computador diz-se **Arquitetura Centralizada**, quando vários computadores realizam a mesma funcionalidade diz-se **Arquitetura Descentralizada**, caso exista uma combinação entre as duas, é uma **Arquitetura Híbrida** [28]. Assim, as arquiteturas de sistema centralizadas caracterizam-se por ter um ponto central de execução e controlo, enquanto que as descentralizadas se caracterizam por ter múltiplos locais onde são executadas as mesmas funções ou similares de controlo, processamento ou transformações [27].

Uma **Arquitetura Distribuída** é caracterizada por dois ou mais processadores autónomos conectados por uma interface de comunicação que executa um sistema distribuído [29].

Os principais elementos das arquiteturas podem ser identificados como **Componentes**, que representam unidades modulares com interfaces de pedidos e respostas bem definidas e substituíveis no seu ambiente, e como **Conectores** responsáveis pela interação entre componentes [28]. Os componentes e conectores das arquiteturas podem ser organizados/configurados de diversas formas (estilos de arquitetura), sendo que, para os sistemas distribuídos alguns dos mais importantes são: **Arquiteturas em camadas**, onde os componentes são organizados por camada em que as camadas superiores têm permissão para invocar as camadas inferiores e não ao contrário; **Arquiteturas baseadas em objetos**, onde cada objeto corresponde a uma componente que comunica com as outras através de invocações de métodos remotos; **Arquiteturas baseadas em eventos**, onde os componentes comunicam pela propagação de eventos que

podem conter dados; **Arquiteturas centradas nos dados**, onde os componentes comunicam através de um repositório de dados comum [28].

As arquiteturas para sistemas distribuídos podem ser **Centralizadas** baseadas em modelos cliente-servidor com organização vertical (ou hierárquica) de comunicação e controlo por camadas e separação lógica de funções entre cliente (pedidos) e servidor (respostas); **Descentralizadas** com modelos *peer-to-peer* organização horizontal de comunicação e controlo em que cada *peer* funciona como cliente e servidor; ou **Híbridas** (combinando as cliente-servidor com *peer-to-peer*) [28].

A implementação de sistemas distribuídos pode ser feita com o recurso a diversas tecnologias, como por exemplo: *Common Object Request Broker Architecture* (CORBA); *Service-Oriented Architecture* (SOA); modelos cliente-servidor; ou agentes de software (Agentes).

CORBA é uma norma definida pelo *Object Management Group* (OMG) desenhada para facilitar a comunicação entre sistemas que foram desenvolvidos em diferentes plataformas. A CORBA usa um modelo baseado em objetos, embora os sistemas que a usam não o tenham de ser. [32]

As arquiteturas SOA são arquiteturas que suportam sistemas orientados a serviços, onde estes estão disponíveis numa rede de computadores através de um protocolo de comunicação. Um serviço é uma representação lógica de uma atividade de negócio repetível com um resultado específico, independente, que pode conter outros serviços e em que os consumidores não têm conhecimento de como é executado. [33]

### 2.1.1 Agentes

Um **Agente** é uma entidade, que inserida num determinado ambiente, é capaz de realizar ações flexíveis e autónomas para executar as tarefas para as quais foi projetado [34]. Uma das principais características dos agentes é a sua autonomia, visto que conseguem executar ações sem a direta intervenção de outros agentes, tendo o controlo das suas ações e do seu estado interno [35]. Para

além da autonomia, os agentes também possuem a capacidade de comunicar com outros agentes, de se adaptar, aprender, e reagir a alterações do ambiente [36].

Quando vários agentes partilham um ambiente comum e interagem entre si, podendo essa informação desencadear ações de agentes, diz-se que os agentes fazem parte de uma comunidade de agentes, também chamada de **Sistema Multiagentes (M.A.S – Multi-Agent System)** [37]. Derivado destes sistemas existe outro conceito bastante difundido, **Agentes de Software**, que podem ser definidos como programas de software que comunicam com os seus pares através de mensagens baseadas numa linguagem de comunicação de agentes e que podem ser classificados como agentes de colaboração, de interface, móveis, de informação, de reação, híbridos, heterogéneos e inteligentes [38],[39].

Existem diferentes formas de comunicação/colaboração entre agentes: coordenação direta, em que os agentes fazem a sua própria coordenação; e coordenação assistida, onde os agentes recorrem a sistemas para fazer a coordenação entre si [39]. Ao sistema utilizado na coordenação assistida, ilustrado na Figura 2.2, dá-se o nome de **Sistema Federado**.

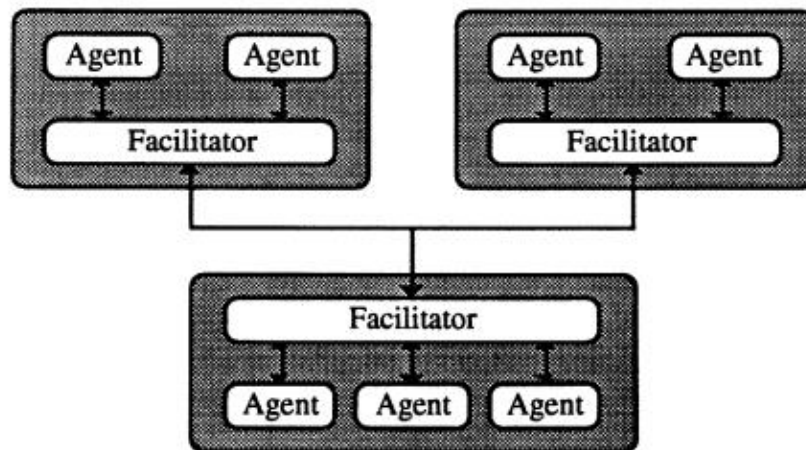


Figura 2.2 – Sistema Federado [39]

Existem diversas normas que permitem definir regras na comunicação entre agentes, minimizando erros e permitindo a interação entre diferentes sistemas utilizando uma linguagem em comum.

Uma destas normas é FIPA<sup>4</sup>, uma norma da *IEEE Computer Society*, que define um conjunto de especificações usadas na definição e implementação de sistemas multiagentes. Entre as muitas especificações descritas pela FIPA, está *Agent Interaction Protocol Suite* (AIPS) que define um conjunto de normas para a interação/comunicação entre agentes. Este conjunto de normas descreve uma linguagem (*Agent Communication Language* - ACL) que define uma semântica e uma sintaxe para a comunicação entre agentes. [40]

Communicative Act (CA)	Base CA	Assertive	Commissive	Directive	Mediate	Phatic	Query
<i>accept-proposal</i>	inform			X			
<i>agree</i>	inform		X				
<i>cancel</i>	disconfirm					X	
<i>cfp</i>	query-ref			X			
<i>confirm</i>	confirm	X					
<i>disconfirm</i>	disconfirm	X					
<i>failure</i>	inform					X	
<i>inform</i>	inform	X					
<i>inform-if</i>	inform	X					
<i>inform-ref</i>	inform	X					
<i>not-understood</i>	inform					X	
<i>propagate</i>	inform				X		
<i>propose</i>	inform			X			
Proxy	inform				X		
<i>query-if</i>	request						X
<i>query-ref</i>	request						X
<i>refuse</i>	disconfirm; inform					X	
<i>reject-proposal</i>	inform			X			
<i>request</i>	request			X			
<i>request-when</i>	inform			X			
<i>request-whenever</i>	inform			X			
<i>subscribe</i>	Request -whenever						X

**Figura 2.3 –FIPA Communication Acts [40]**

O AIPS é composto por diferentes protocolos semânticos: *Interaction Process* (IP), *Communicative Acts* (CA), *Content Logic*, e *Content Ontologies*. As CA, ilustradas na Figura 2.3, definem um conjunto de ações que podem ser executa-

---

<sup>4</sup> <http://www.fipa.org/>



das pelos agentes. Os IP, descritos na Figura 2.4 , permitem o uso das CA como parte das mensagens ou *workflows* que suportam a troca de informação ou atribuição de tarefas entre os agentes. O *Content Logic* e o *Content Ontologies* contêm o detalhe da ação a executar e a que contexto se refere. [40]

Interaction Protocol	Task/ Info-sharing	Push/ Pull	1-1/ 1-m Receivers	Other Features
Request	Task	Pull	1-1	Cancelable (by initiator)
Request-when(ever)	Task	Push	1-1	Cancelable
Query	Info	Pull	1-1	Cancelable
Contract-Net/Iterated CN	Task	Push	1-m	Cancelable, iterated version is a multi-round IP
English/ Dutch Auction	Info	Pull	1-m	Cancelable
Broker	Info	Pull	1-m	Cancelable
Recruit	Task	Pull	1-1	Cancelable
Subscribe	Info	Push	1-1	Not cancelable
Propose	Task	Pull	1-1	Not cancelable

Figura 2.4 – FIPA Interaction Protocols [40]

Outro exemplo de normas para a comunicação entre agentes é a KQML que usa também um conjunto de ações, tais como *replay*, *tell*, *deny*, etc, nas mensagens juntamente com dados adicionais escritos em vários *slots* (como *content*, *in-reply-to*, *sender*, *receiver*, etc) [41].

Para além destas duas existem mais exemplos tais como: ARTIMIS *Communication Language* (ARCOL); *Knowledge Interchange Format* (KIF); *Domain independent COOrdination Language* (COOL) [42].

## 2.2 Constituição dos Sistemas de Vigilância

Tendo em conta as definições de vigilância (início do Capítulo 2) e de sistema descritas anteriormente, pode-se dizer que um **Sistema de Vigilância**, na sua generalidade, é um conjunto de componentes de hardware (normalmente câmaras e outros tipos de sensores) que enviam a informação para um conjunto de componentes de software que permite a deteção e alerta de eventos. Sistema de Vigilância também pode ser definido como uma ferramenta tecnológica que auxilia os humanos, oferecendo uma perceção e capacidade de raciocínio alargada sobre os locais que estão a ser monitorizados [14],[43].

Os Sistemas de Vigilância são tipicamente compostos por cinco grandes blocos: Deteção de Objetos, Reconhecimento de Objetos, Seguimento, Análise comportamental e Base de Dados [44], sendo muitas das vezes, nas ferramentas usadas dentro de cada um destes blocos, que reside a diferença entre os sistemas.



Figura 2.5 – Blocos que compõem tradicionalmente os Sistemas de Vigilância usando a visão

Na **Deteção de Objetos** podem ser usados vários sistemas, como câmaras de vídeo com visão computacional (construção de descrições explícitas de objetos físicos a partir de imagens [45]), utilização de sensores LiDAR, câmaras TOF (*Time of flight*), sensores LiDAR 3D, câmaras de infravermelhos, sensores de ultrassons ou sensores de infravermelhos, câmaras multiespectrais [46].

Dependendo do sistema usado, podem ser aplicadas diferentes técnicas para a detecção de objetos, como por exemplo os algoritmos apresentados por Zhang et al. [47], Harmeling et al. [48], Robert T. Collins et al.[49], O. Javed and M. Shah [50], V. Nair and J. J. Clark [51], W. Niu et al. [52].

Também, dependendo dos sistemas utilizados, podem ser usados diferentes métodos para o **Reconhecimento de Objetos**, os baseados em modelos 2D (ex.: [53]), com ou sem modelos de formatos, e os baseados em modelos 3D (ex.: [54], [55]) [1]. Interligado com a identificação de objetos está o **Seguimento**, cujo objetivo é seguir um dado objeto, identificando a sua posição nas sucessivas leituras.

Posteriormente, a **Análise de Eventos** permite detetar situações de risco através da análise dos objetos detetados e do seu comportamento. Para a análise de eventos existem várias abordagens, tais como o DTW (*Dynamic Time Warping*), algoritmo para a medição de semelhanças entre duas sequências que podem variar em tempo e velocidade, ou HMM (*Hidden Markov Models*), modelo estatístico em que o sistema modelado é assumido com um processo de *Markov* com parâmetros desconhecidos [44].

Por último a **Base de Dados**, onde é guardada toda a informação recolhida, e sobre a qual pode haver pesquisa ou análise de dados. Nesta área existem alguns trabalhos realizados no sentido de criar modelos de dados para suporte dos diferentes níveis de abstração dos dados e outros no sentido de analisar e extrair informação da base de dados. Exemplos disso são as referências bibliográficas referidas em [56] e em [57].

## 2.3 *Arquiteturas de Sistemas de Vigilância*

A Arquitetura de um Sistema de Vigilância descreve os vários elementos que o constituem, como por exemplo, a aquisição de dados ou o processamento da informação, e a interligação entre estes elementos.

Em relação aos Sistemas de Vigilância, foram estudadas as arquiteturas centralizadas e as arquiteturas distribuídas.

### 2.3.1 **Arquitetura centralizada**

Como referido em 2.1, uma arquitetura centralizada tem por base um ponto central para onde converge toda a informação. Um exemplo deste tipo de arquitetura, em sistemas de vigilância, é o tradicional monitor que podemos ver em supermercados, lojas, etc., onde é possível ver as imagens das várias câmaras que compõem o sistema.

Este tipo de arquiteturas, embora ainda muito usadas em locais de pequenas dimensões e com poucas câmaras, têm algumas desvantagens, sendo a principal o facto de estar todo o sistema dependente de apenas um ponto para o seu funcionamento. No entanto, no mercado ainda é uma das arquiteturas mais usadas, sendo uma das razões o seu custo de implementação ser mais reduzido, visto que usa tecnologia mais barata e apenas requer uma unidade de processamento.

Porém, as arquiteturas centralizadas têm vindo a sofrer algumas alterações que permitem melhorar o seu desempenho, como por exemplo a desagregação do sistema central em módulos. Um dos exemplos desta evolução é o Sistema de Vigilância proposto pelo PRISMATICA (*PRoactive Integrated systems for Security Management by Technological, Institutional and Communication Assistance*) [58], que se caracteriza por um sistema multisensor com distribuição de processamento dos vários sensores, que posteriormente enviam a informação para uma unidade de processamento central onde é feita a interpretação da informação reunida. A arquitetura deste sistema é ilustrada na Figura 2.6, onde é possível ver que todo o sistema é dependente da unidade central MIFSA [19].

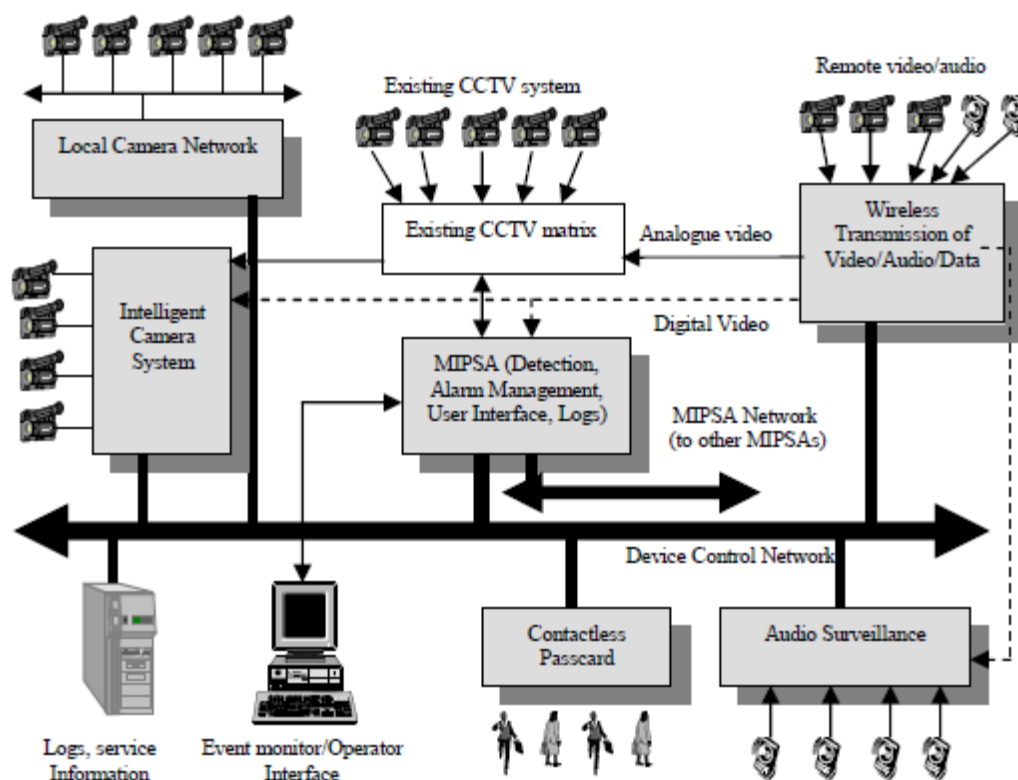


Figura 2.6 – Arquitetura do sistema PRISMATICA [19]

O ADVISOR (*Annotated Digital Video for Intelligent Surveillance and Optimised Retrieval*) [59], que foi projetado para ser usado sobretudo em estações de metro, permitindo detetar situações de perigo (como acidentes, violência ou vandalismo). Este pode ser considerado um sistema semi-distribuído, uma vez que é constituído por vários subsistemas que distribuem o processamento de todo o sistema. No entanto, cada nó é mais um exemplo de uma arquitetura centralizada.

Este sistema é constituído por uma unidade central de processamento que agrupa vários computadores responsáveis pelo processamento da informação recebida pelas câmaras. Este processamento é distribuído pelos vários computadores, sendo a informação trocada segundo normas internacionais, como por exemplo, o uso Baseline JPEG (ISO/IEC IS 10918-1) para a comunicação de dados de vídeo. A arquitetura seguida por este sistema é ilustrada na Figura 2.7, onde são identificados os vários blocos constituintes da unidade de processamento [59].

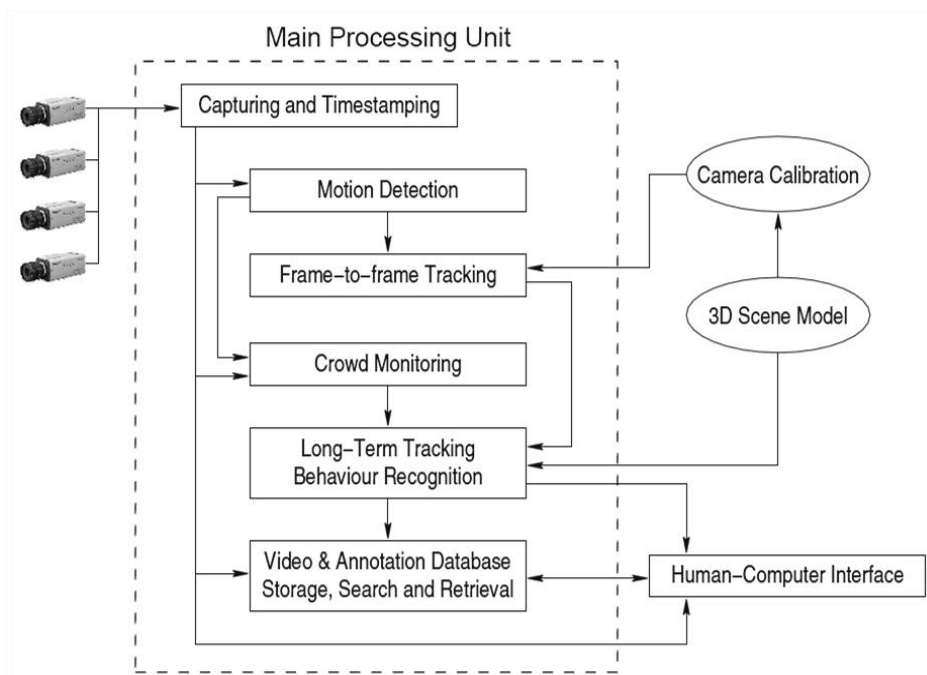


Figura 2.7 – Arquitetura do sistema ADVISOR [59]

Outra solução semi-distribuída, a AMFIS (*Aufklärung mit Miniaturfluggeräten im Sensorverbund*), propõe um sistema constituído por uma estação de controlo universal, um conjunto de sensores e veículos de transporte de sensores, que comunicam entre si através de mensagens XML [60]. O uso das mensagens em XML facilita a integração com outros sistemas, sendo uma linguagem facilmente implementável em diversos tipos de linguagem de programação.

Este sistema é baseado na arquitetura ilustrada na Figura 2.8, sendo que o “AMFIS Connector” corresponde à estação de controlo e os sensores estão incluídos no “Sensor Carrier Control Application”. O “User Interface Application” corresponde às interfaces disponibilizadas aos vários clientes da aplicação [60].

O conector permite a integração de novos elementos no sistema através de uma linguagem comum utilizada para a comunicação entre os diferentes módulos. Esta estação de controlo é responsável pela gestão de tarefas, planeamento de missões, controlo de sensores e plataformas móveis, alerta de eventos, fusão e exploração de dados, geração de alarmes, relatórios e armazenamento.

Para além destas tarefas, o AMFIS conector envia a todos os clientes conectados todas as mensagens recebidas, sendo depois estes que decidem se devem tratar e como cada mensagem. Os clientes quando se conectam recebem primeiro a descrição detalhada do sensor e a partir daí todos os dados que o sensor vai obtendo ciclicamente.

Por outro lado, os sensores recolhem a informação sensorial, estática ou móvel (utilizando veículos comandados pela estação de controlo) e enviam os dados para o sistema.

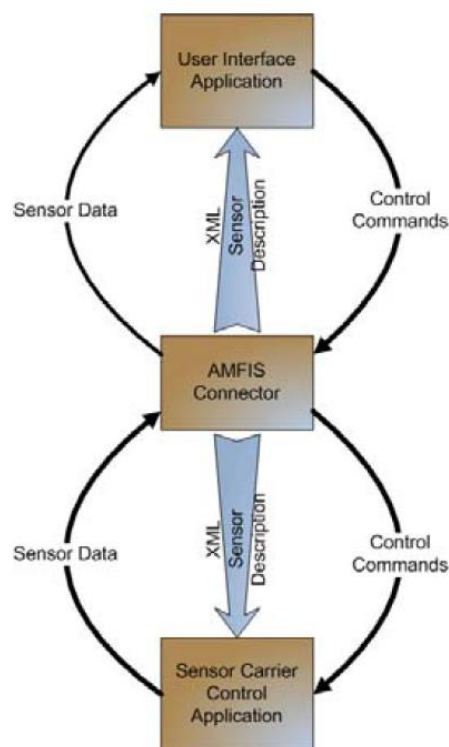


Figura 2.8 – Arquitetura do sistema AMFIS [60]

Mais recentemente, Mei Kuan Lime et al. apresentam em [61] uma arquitetura dividida em três camadas, a camada sensorial - onde é feita a aquisição e processamento do vídeo; parte de análise e lógica - onde é construída a base de conhecimento do ambiente, seguido da análise e lógica e usando o Princípio da

Composicionalidade<sup>5</sup>; e parte utilizador – onde é apresentado o alarme de cada evento.

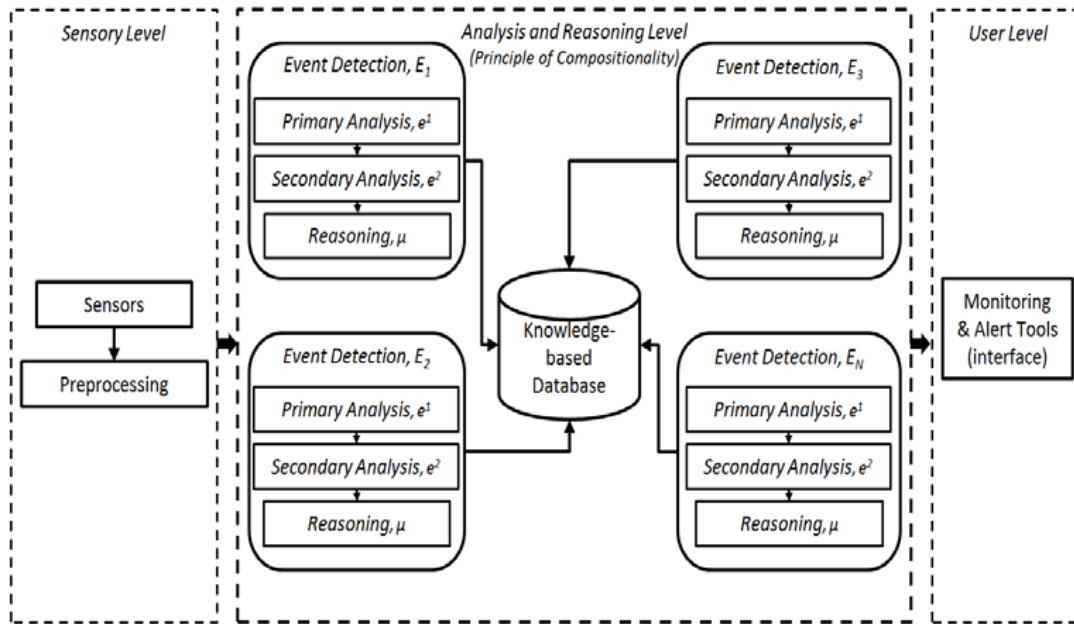


Figura 2.9 – Arquitetura do sistema apresentado em [61]

### 2.3.2 Arquitetura distribuída

Com a evolução tecnológica, com o objetivo de reduzir a dependência do sistema a um só ponto e com o aumento da capacidade computacional, têm vindo a ser cada vez mais abordadas propostas de arquiteturas distribuídas para os sistemas de vigilância. Estes sistemas permitem fracionar o processamento da informação, isto é, são criados nós responsáveis pelo processamento da informação de um grupo de sensores ou em alguns casos cada câmara/sensor pode ser uma unidade de processamento(nó).

As arquiteturas multiagentes têm vindo a ser cada vez mais usadas em Sistemas de Vigilância distribuídos, havendo alguns exemplos na literatura, como o referido em [4] e [62], onde é proposta uma arquitetura baseada em

<sup>5</sup> O significado de uma expressão complexa é uma função dos significados das suas partes e da forma como estas estão combinadas / o significado de uma frase é determinado pelos significados dos seus constituintes e pelo modo como estes estão combinados" [94]



multiagentes. Neste caso é proposta a subdivisão do cenário em áreas por um conjunto de agentes, sendo cada um deles responsável por analisar uma determinada área.

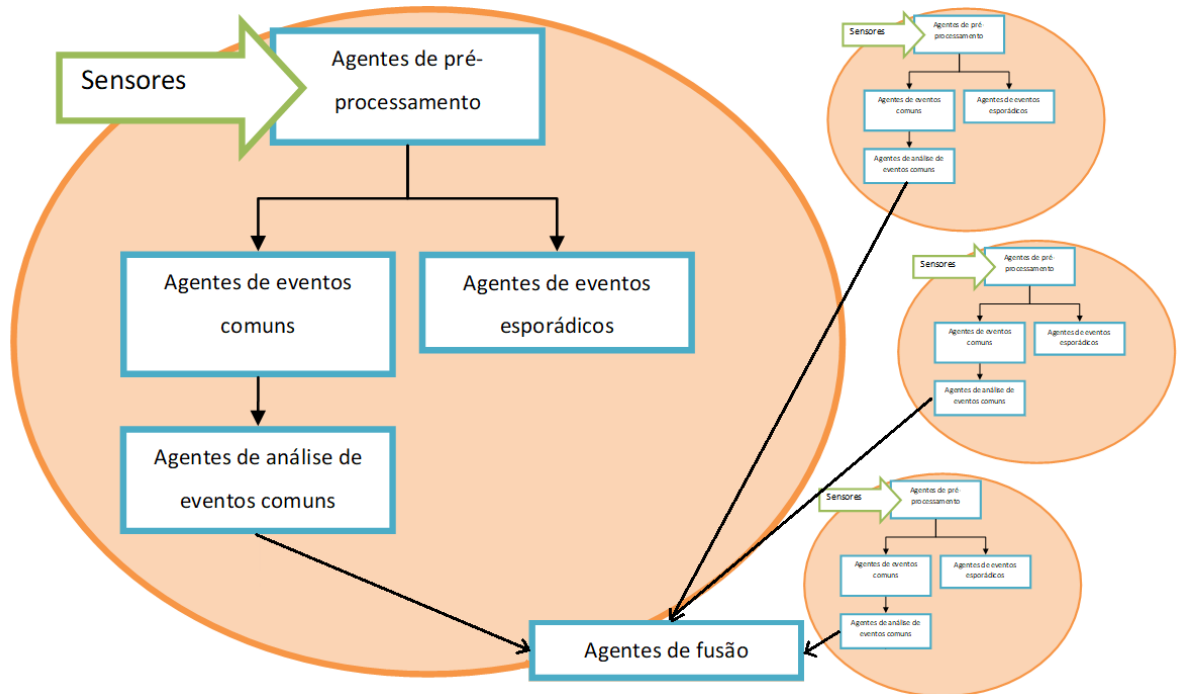


Figura 2.10 – Arquitetura baseada em agentes para monitorização descrita em [4]

Cada conjunto (responsável por uma área), representado na Figura 2.10, é composto por agentes de pré-processamento da informação recebida pelos sensores, agentes que analisam os eventos comuns, agentes que analisam os eventos esporádicos e agentes que agrupam o conhecimento proveniente dos agentes que analisam os eventos comuns. Existem depois os agentes de fusão que recebem a informação dos vários conjuntos e a integram numa nova base de conhecimento comum.

Em “*Multi-agent Based Distributed Semi-automatic Sensors Surveillance System Architecture*” [63] é descrita uma outra arquitetura baseada em multiagentes agrupada em três níveis: dispositivos, processamento de sensores e assistência/suporte. Esta arquitetura descentralizada, ilustrada na Figura 2.11, faz uma abordagem aos sistemas de vigilância de uma forma distribuída, subdividindo as tarefas pelos vários agentes existentes.

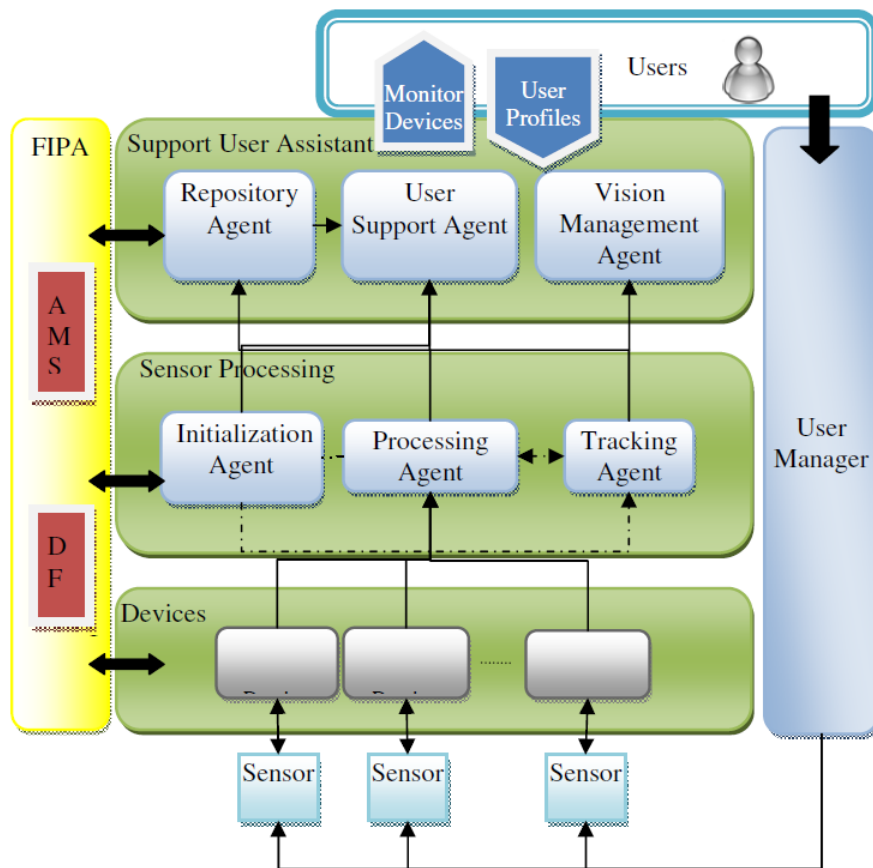


Figura 2.11 – Arquitetura multiagentes [63]

Outro exemplo deste tipo de arquiteturas como o descrito em “*Multi-Agent Framework in Visual Sensor Networks*” [64], onde os autores desenvolveram um sistema em que cada agente decide executar uma tarefa tendo em conta os outros agentes e a informação presente na rede. Cada câmara é representada e controlada por um único agente, que apenas tem conhecimento de alguns eventos externos e que toma decisões tendo em conta estas limitações.

Mais recentemente foram apresentadas em “*Multi-agent systems for protecting critical infrastructures: A survey*” [65] várias aplicações/soluções de sistemas multiagentes para diversas áreas, como comércio eletrónico, *e-health*, detecção de

intrusões em rede informáticas, Telemática<sup>6</sup> e transportes e monitorização de espaços.

Em relação à monitorização de espaços, é apresentada uma solução proposta por Abielmona et al, em "*Multi-Agent System Information Fusion for Environment Monitoring*" [66] onde é descrita uma nova abordagem baseada numa técnica chamada de *tree-in-motion mapping* (TIMM), onde é feita a fusão dos dados através da redução da entropia. Com os dados baseados numa estrutura Quadtree<sup>7</sup>, esta abordagem permite apresentar uma visão multidimensional e multi-resolução do ambiente monitorado.

Neste trabalho é também explorada a forma como lidar com a fusão de dados multisensor (MSF - *multi-sensor fusion*), podendo ser feito o processamento centralizado ou distribuído. No C-MSF (*Centralized MSF*) todos os sensores enviam a informação local para o processador que faz a fusão dos dados e faz uma estimativa global. No D-MSF (*Decentralized MSF*), os sensores enviam a informação local para os outros sensores que processam a informação localmente e, juntos, sintetizam uma estimativa global de todas as estimativas locais dadas por cada sensor [66].

Outra abordagem de sistemas distribuídos usada em sistemas de vigilância são as chamadas Câmaras Inteligentes. Ahmed Nabil Belbachir, em "*Smart Cameras*" [67] define câmaras inteligentes como sistemas de visão que para além de captar imagens, são capazes de extrair informação das imagens captadas, podendo esta informação gerar eventos ou ações usadas em sistemas inteligentes automáticos [67]. Algumas das abordagens seguidas para arquiteturas distribuídas, baseiam-se no uso de conjuntos destas câmaras, onde são definidos meios de cooperação entre elas.

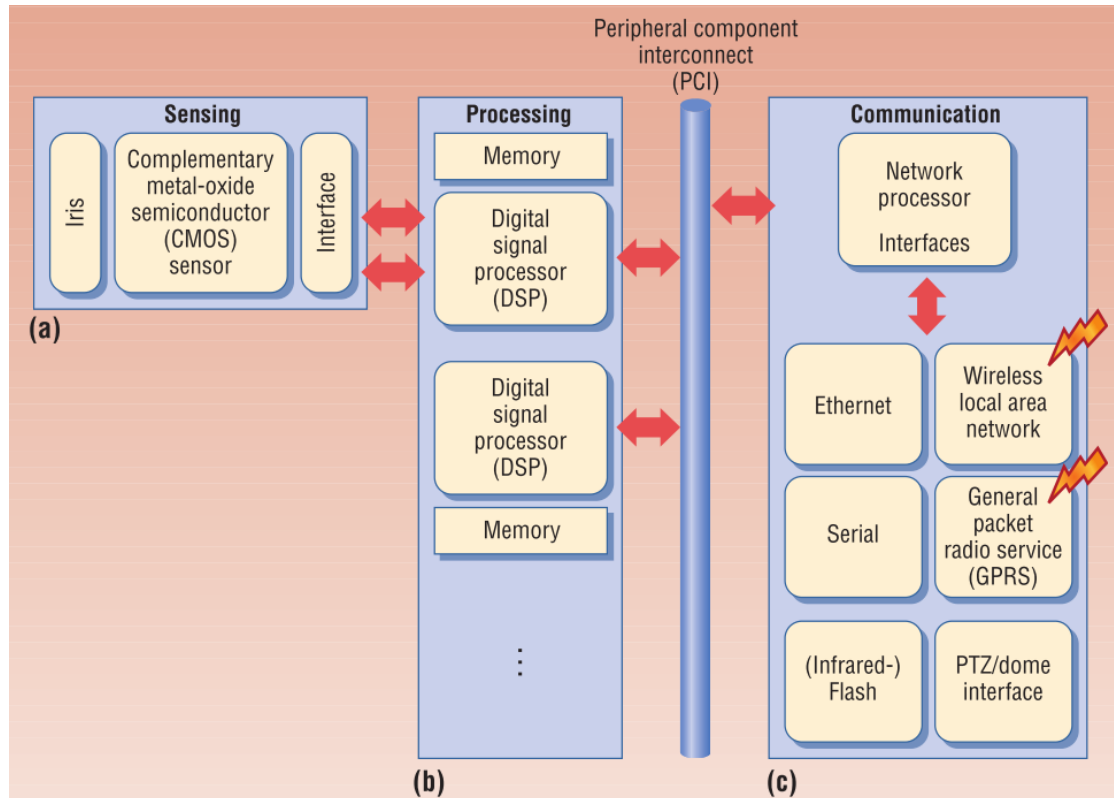
---

<sup>6</sup> Telemática é a comunicação à distância de um ou mais conjunto de serviços informáticos fornecidos através de uma rede de telecomunicações, <http://www.priberam.pt/dlpo/telem%C3%A1tica>

<sup>7</sup> Estrutura de dados em árvore onde cada nó interno tem exatamente quatro filhos [95]

Em [68] é descrito o uso de câmaras inteligentes como base para uma arquitetura de um sistema distribuído. As câmaras inteligentes propostas neste sistema são baseadas numa arquitetura para o hardware e uma para o software.

Na Figura 2.12 é ilustrada arquitetura para o hardware, que permite a recolha, processamento e comunicação da informação obtida pelo sensor CMOS (*Complementary metal-oxide semiconductor*).



a) unidade sensorial b) unidade de processamento c) unidade de comunicação

**Figura 2.12 – Arquitetura do Hardware.** [47]

Na Figura 2.13 está ilustrada a arquitetura para o software, que aproveitando as capacidades de comunicação e o uso de agentes de software móveis, permite a distribuição de tarefas pelas várias câmaras que constituem o sistema

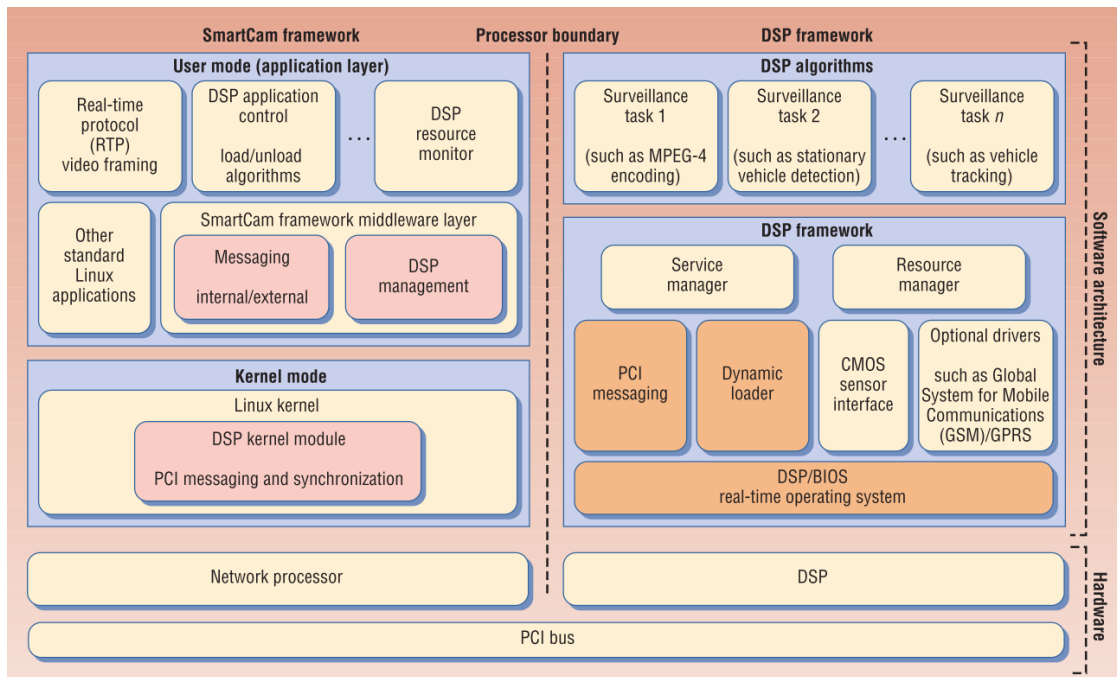


Figura 2.13 – Arquitetura do software das câmaras inteligentes [68]

Outro exemplo de arquitetura para sistemas de vigilância baseados em multisensores, é apresentado em [69], onde é proposta uma rede de sensores inteligentes capazes de realizar tarefas de detecção de objetos em tempo real, seguimento e classificação. Para além da capacidade de processamento, detecção de objetos e seguimento distribuído, o sistema descrito em [69], permite a fusão da informação obtida pelos vários sensores a nível central, melhorando a capacidade de detecção dos vários tipos de eventos.

Para além das arquiteturas descritas existem mais exemplos na literatura, muitos deles descritos em *"Intelligent multi-camera video surveillance: A review"*, onde é feita uma revisão de sistemas inteligentes de vídeo vigilância - sistemas capazes de extrair informação de uma grande quantidade de vídeos, obtidos por câmaras de vigilância e automaticamente detetar, seguir e reconhecer objetos e identificar e analisar a sua atividade [3].

Para além dos sistemas descritos em *“Intelligent multi-camera video surveillance: A review”*, usado como referência em várias publicações nesta área, têm vindo a ser publicadas mais recentemente várias novas abordagens, como as descritas em:

- *“Intelligent video surveillance system architecture for abnormal activity detection”* [70] que descreve um sistema de monitorização baseado em análise (algoritmos de deteção e seguimento de objetos) de vídeos obtidos por diversas câmaras;

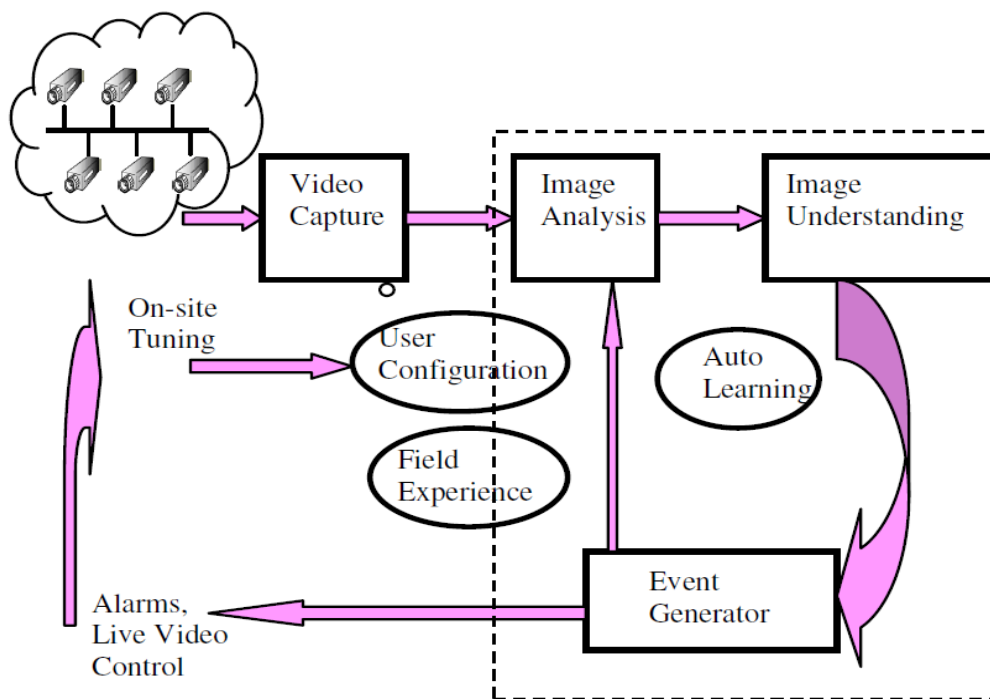
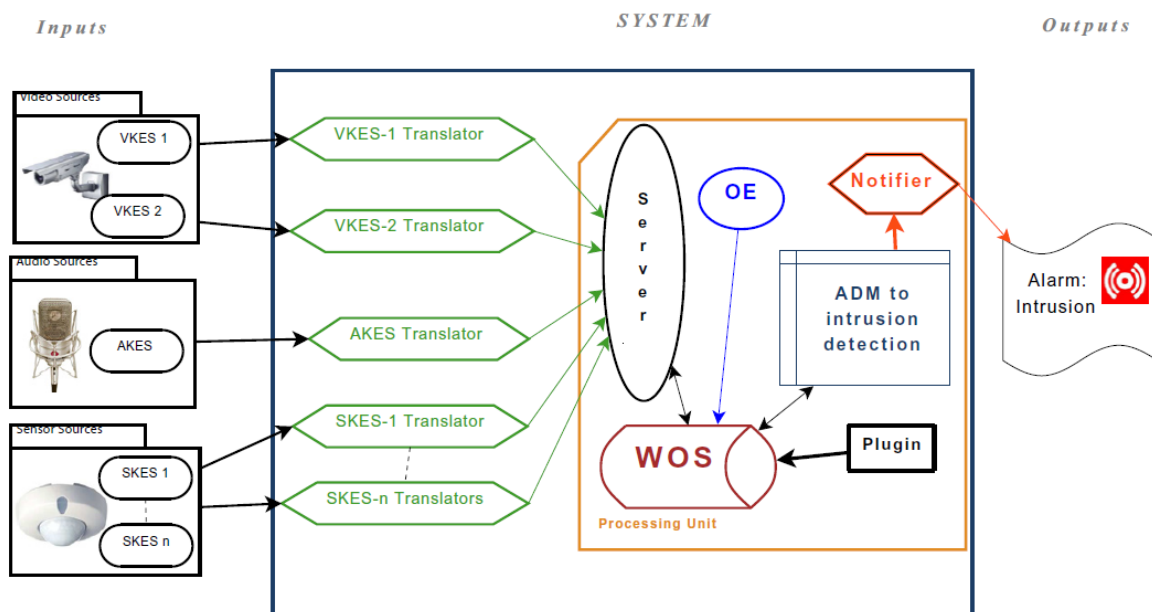


Figura 2.14 – Arquitetura do sistema apresentado em [70]

- *“Intelligent surveillance system with integration of heterogeneous information for intrusion detection”* [71] onde é proposto um sistema multisensor baseado numa ontologia que permite a recolha de informação de diferentes fontes (vídeo, áudio, e outros sensores) e a sua integração num sistema de deteção baseado em regras;



KES - Knowledge Extraction Systems (V-Video, A-Audio, S-Sound); WOS - Warehouse of the Objects of the Scenario; ADM – Alarm Detection Module; OE- Object Eliminator;

**Figura 2.15 – Arquitetura do sistema apresentado em [71]**

- "Developing intelligent surveillance systems with an agent platform" [72] e "An agent-based approach to understand events in surveillance environments" [73] é apresentada uma arquitetura multiagentes inteligente, que usando uma base de conhecimento (definida através do conhecimento dos humanos de situações anteriores ou com base em eventos anteriores registrados pelo sistema), utiliza um modelo baseado na análise de comportamento, sendo que, cada análise representa uma tarefa que é realizada por um agente.
- Também baseado na mesma plataforma, OCULUS, é apresentado em "Dynamic weighted aggregation for normality analysis in intelligent surveillance systems" [74] uma revisão do trabalho realizado pelos autores em [4] e descrita uma nova abordagem para detecção de eventos. Esta abordagem atribui peso aos eventos detetados, usando-os na detecção de próximos eventos.

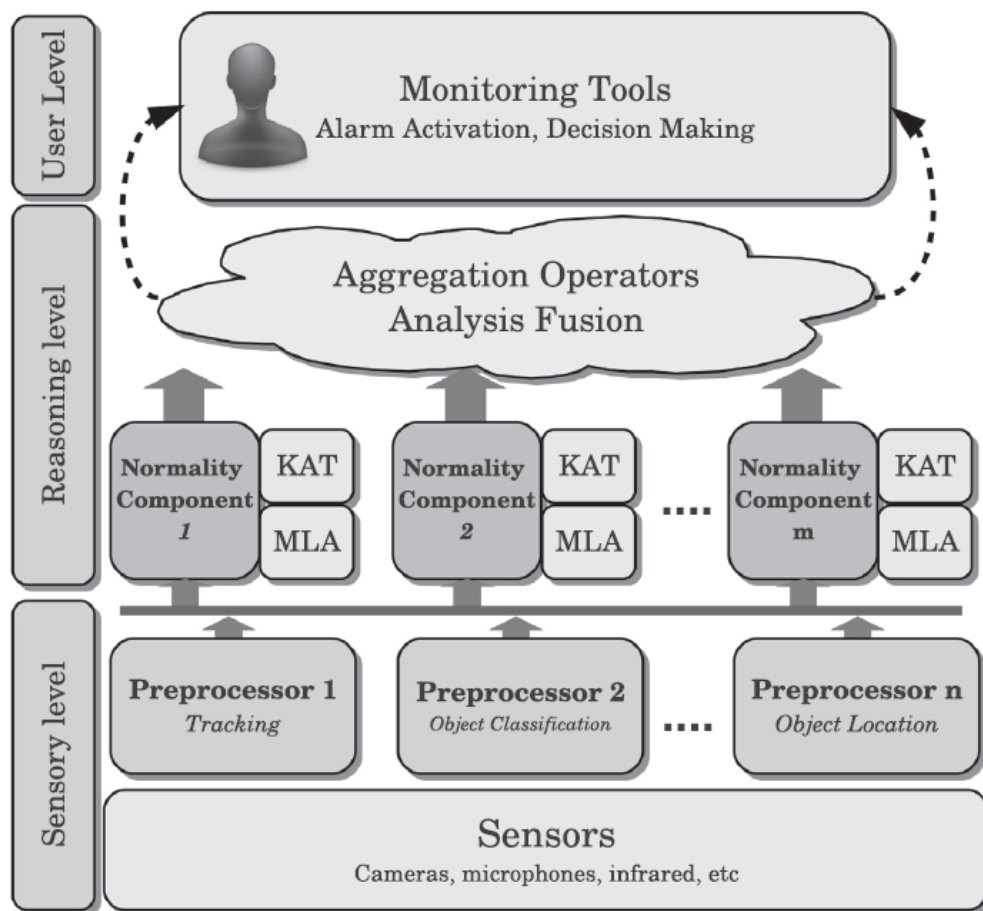


Figura 2.16 – Arquitetura do sistema OCULUS [74]

## 2.4 Análise comparativa

Atualmente, os sistemas de vigilância denotam uma fraca cooperação entre os seres humanos e os sistemas de vigilância. A "Cooperação" neste momento é basicamente o fornecimento de informações pelos dispositivos de vigilância para os seres humanos. Há "pouca ou nenhuma" cooperação entre os dispositivos dentro dos atuais sistemas de monitorização e detecção de eventos de vigilância, e não existe uma coordenação de seres humanos para reagir a eventos, com base na localização geográfica dos seres humanos e eventos [3].

O tipo de evento e a utilização da localização, quer dos eventos quer dos recursos (humanos), permite diminuir o tempo de resposta aos eventos, a adequação do tipo de resposta tendo em conta o tipo de evento e gerir os recursos conforme a sua localização.



Tem existido um esforço da comunidade científica em fazer evoluir estes sistemas, no entanto, a maioria dos sistemas em uso continuam a ser centralizados e de difícil reconfiguração. Em alguns casos, a constituição da rede de câmaras ou sensores só é possível quando se está ainda na definição inicial do sistema.

Outra das áreas ainda com algumas lacunas é a correlação de informação obtida por múltiplos sensores, como por exemplo a definição de padrões que permitam interligar a presença de um gás com o aumento da temperatura [71].

As novas propostas apresentadas nos últimos anos, de arquiteturas de sistemas de vigilância distribuídos, permitem a abordagem ao problema subdividindo o processamento por diferentes unidades e minimizar os pontos de falhas gerais do sistema e aumentar a capacidade de processamento, no entanto, ainda existem pontos que necessitam de ser melhorados [1], [4], como é o caso de:

- Adaptação, fusão de dados e métodos de seguimento em ambientes cooperativos e com múltiplos sensores;
- Melhores técnicas de identificação e classificação de objetos;
- Novos protocolos e modelos de metadados de comunicação e integração entre diferentes módulos;
- Desenvolvimento de arquiteturas com algoritmos de processamento de dados multisensor e capacidade reconfiguração em tempo real;
- Arquiteturas adaptáveis a diferentes tipos de ambientes;

De uma forma genérica existem vários pontos onde existe a necessidade de melhorar, tais como:

- Diminuição do tempo de resposta a eventos;

- Adequação do tipo de resposta aos vários tipos de eventos;
- Redução de falsos alarmes;
- Redução de custos;
- Diminuição da dependência dos sistemas dos humanos.

## Metodologia de Investigação

A metodologia de investigação usada como base para o desenvolvimento deste trabalho foi baseada na metodologia clássica de investigação, representada na Figura 3.1.

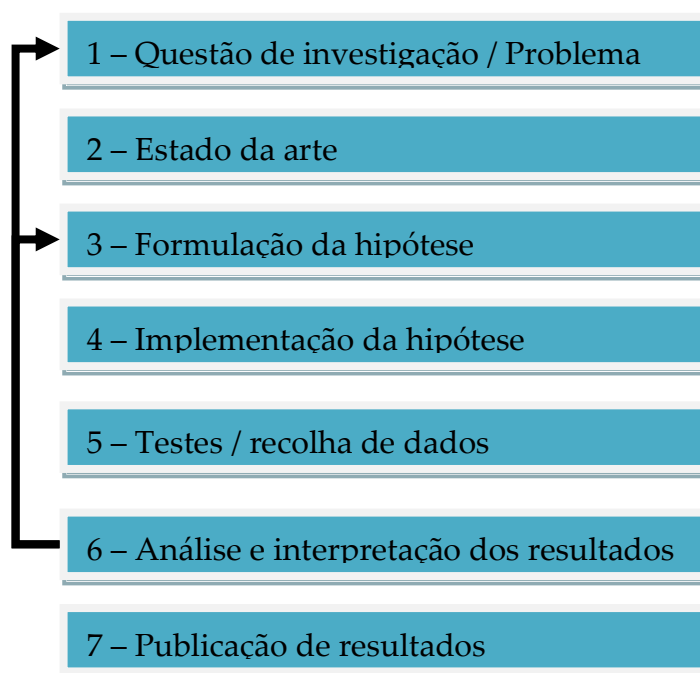


Figura 3.1 – Metodologia de investigação adaptado de [75]

Com base nesta metodologia, o trabalho foi iniciado pela análise do problema existente, descrita neste documento na secção 1, o que permitiu a definição de uma questão de investigação, enunciada na secção 1.1.

Após a identificação da questão de investigação foi feito o estudo do estado da arte, descrito no capítulo 2. Com base neste estudo, foi elaborada a hipótese que pretende dar resposta à questão de investigação levantada.

A hipótese descrita na secção 1.2 visa o desenvolvimento de uma arquitetura para Sistemas de Vigilância. Como forma de validação da hipótese colocada foi desenvolvido um Sistema de Vigilância baseado na arquitetura proposta permitindo validar, de uma forma prática, a hipótese colocada.

### 3.1 *Contribuição esperada*

O trabalho teve como objetivo principal o desenvolvimento de uma arquitetura para um Sistema de Vigilância Inteligente baseado em agentes georreferenciados, que cooperem entre si através de uma linguagem comum, conseguindo um comportamento global emergente com vista à sua aplicação na área da Segurança de pessoas e bens.

Com as capacidades e velocidades de comunicação disponíveis hoje em dia no mercado, foi concebido um sistema de monitorização que, tirando partido das elevadas taxas de transmissão de dados, da capacidade de lidar com vários tipos de informação em simultâneo (multimédia), da localização de dispositivos e do suporte a múltiplas plataformas (fixas e móveis), permite disponibilizar serviços de vigilância que contribuem para o aumento da Segurança de pessoas e bens através de uma nova oferta de serviços.

A capacidade de processamento distribuído e geolocalizado (pela utilização de agentes de software) e a velocidade de comunicação entre os elementos componentes do sistema permite que sejam despoletadas ações em tempo real, com base em regras, para reagir a determinados eventos como ações de vandalismo, roubo ou intrusão, ou eventos disruptivos como incêndios, fugas de gases, terremotos, inundações ou falhas em sistemas - iminentes ou em curso.

A arquitetura proposta permite facilitar um modelo reativo de atuação, que visa impedir o início ou a continuação de atividades comprometedoras da segurança de pessoas ou bens ou, noutros casos, de reagir atempadamente de

forma a minorar os efeitos adversos de eventos disruptivos. Este modelo pode incorporar um grande número de variáveis, reagindo em função de regras que permitem a adaptabilidade do sistema.

As principais metas estabelecidas para este trabalho foram:

- O desenvolvimento de modelos de cooperação entre agentes (dispositivos hardware que cooperam entre si e com humanos) considerando a georreferenciação de eventos e recursos;
- A definição do modelo inteligente de inferência, utilizando métodos especializados e técnicas de inteligência artificial para lidar com imprecisão e incerteza tanto na combinação das variáveis, como na execução das regras (implicação) e ainda na agregação das mesmas para chegar a uma conclusão;
- A definição de regras de comunicação entre agentes, nomeadamente, na construção de modelos de mensagens e validação das mensagens enviadas/recebidas;
- A integração, adaptação e desenvolvimento de algoritmos de processamento de dados para deteção de eventos específicos como catástrofes naturais, ações de vandalismo ou de intrusão com base em informação recolhida em sensores (imagens, sons, análise de gases, temperatura, etc.);

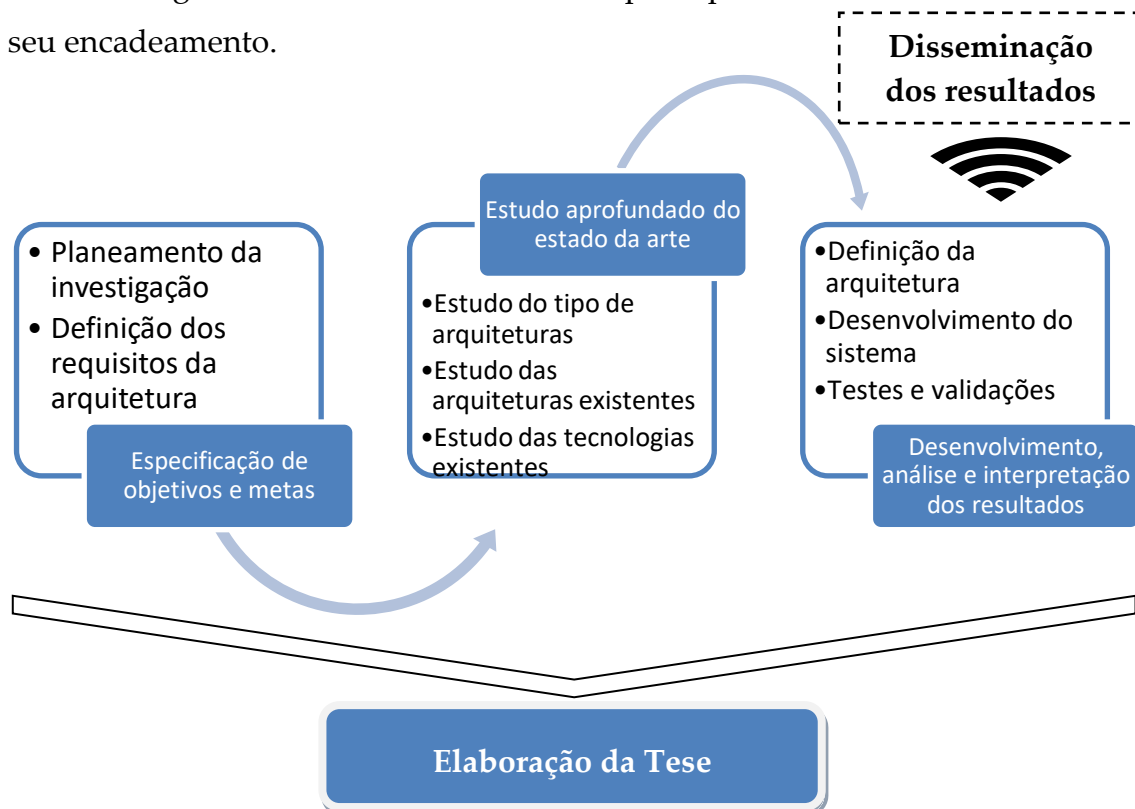
Como resultados diretamente ligados ao trabalho desenvolvido, espera-se:

- Melhorar a integração entre humanos e máquinas;
- A redução no tempo de reação a eventos;
- A descentralização dos Sistemas de Vigilância.

Este sistema permitirá contribuir para a redução dos custos de manutenção de instalações, por exemplo, aumentando o leque de situações em que pode haver uma reação rápida e automática a situações anómalas, minorando os efeitos adversos tanto em aspetos financeiros como de consequências em termos humanos. Por outro lado, o aumento da quantidade de informação processada possibilita o aumento da qualidade na decisão, reduzindo também os custos associados a situações de falsos alarmes. O sistema proposto está desenhado de forma a ser facilmente escalável e adaptável a diferentes tipos de informação sensorial.

### 3.2 *Plano de trabalho*

O desenvolvimento desta dissertação seguiu um plano de trabalho detalhado na Figura 3.2, onde são descritas as principais atividades, assim como o seu encadeamento.



**Figura 3.2** – Plano de trabalho

O trabalho desenvolvido no âmbito deste doutoramento foi parcialmente inserido no trabalho desenvolvido no seio do projeto QREN – "DVA – Sistema Avançado de Vigilância baseado em agentes" do programa do Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico (SI I&DT). Este projeto visava o desenvolvimento de um sistema para suportar uma abordagem colaborativa entre humanos e máquinas na temática da vigilância.

A primeira fase do trabalho, passou pelo estudo detalhado das necessidades dos utilizadores finais de sistemas de vigilância, identificando os problemas que existem nos sistemas atuais. Paralelamente à definição da questão de investigação e à colocação da hipótese, foram identificados os requisitos de utilizador e de sistema como complemento. Nesta identificação e definição de requisitos foram estabelecidos contactos, no âmbito do projeto DVA, com um conjunto de especialistas de entidades representativas na área da segurança, como Prosegur Tecnologia, Prosegur Vigilância Activa, o Dr. Mendes Lopes (ex-Chefe de Divisão do SEF e representante do Governo Português para os acordos de Schengen e Eurodac), a Polícia de Guarda de Fronteiras de Angola, a PT Inovação, a Oni, a Alcatel, e Siemens.

Tendo em conta estes requisitos e a hipótese formulada foi definido um cenário base e um piloto a implementar, que serviu como base de validação do trabalho desenvolvido. Tanto o cenário como o piloto propostos foram definidos tendo em conta a sua polivalência, permitindo validar as vantagens introduzidas pela arquitetura proposta.

Com base no levantamento efetuado no estado da arte e nos resultados do levantamento de requisitos, foi desenhada na segunda fase deste trabalho a arquitetura do sistema proposta na hipótese e descrita detalhadamente no capítulo seguinte.

Após a definição da arquitetura, foram analisadas detalhadamente alternativas às plataformas de desenvolvimento de agentes com vista à identificação de características que possam beneficiar o presente trabalho (e.g. mobilidade de

agentes entre plataformas) e especificada a arquitetura e funcionalidades dos módulos de agentes de software.

A arquitetura desenvolvida teve também em conta o desenvolvimento das componentes apoiadas em primitivas de localização, que permitem criar uma comunidade de sensores georreferenciados mais adequada às funções a implementar.

Paralelamente e considerando os requisitos de processamento de informação sensorial identificados no levantamento de requisitos, foram identificados algoritmos que sirvam os objetivos propostos a nível de deteção de eventos. Estes foram adaptados ao hardware/software projetado e ao sistema de agentes.

Para permitir a validação da arquitetura, foi desenvolvido um sistema baseado em agentes (componentes de software), responsável pela comunicação e interação entre os vários agentes e pela criação de uma comunidade de agentes compreendendo os vários tipos de intervenientes:

- Dispositivos de Vigilância fixos, tipicamente câmaras e sensores amovíveis;
- Dispositivos de Vigilância móveis, por exemplo, humanos com dispositivos móveis ou robôs móveis.

Salientar que no que concerne aos dispositivos móveis foi desenvolvido um agente de software que permite georreferenciar a sua posição. Os dispositivos fixos comunicam com estes, enviando-lhe informação sensorial (multimédia, booleana e real), que permite, por exemplo, no caso de um destes ser um humano, decidir o curso de ação a tomar (ex. anular o alarme, seguir intruso, ler estado, ver imagem).

Por último, foram confrontados os resultados obtidos, usando como base o cenário definido, validando o trabalho desenvolvido.



### 3.3 *Metodologia de validação*

Com vista a validar a hipótese formulada, foi construído um protótipo funcional (composto por software e hardware) para testar e validar a arquitetura proposta. Este protótipo foi dotado de componentes de agente de software equipados com os sensores e câmaras adequados num âmbito definido durante a fase de análise do trabalho. Foram também utilizados dispositivos móveis para demonstração das capacidades de interação com humanos, tendo em conta a sua localização em tempo real.

Na Tabela 3.1, são descritas algumas características/objetivos que serviram como base de análise e validação da arquitetura proposta contra as arquiteturas existentes. As características descritas na tabela são:

- **Escalabilidade do sistema** - Os sistemas existentes hoje em dia são limitados a um certo número de nós, principalmente em arquiteturas centralizadas, e muitos deles não permitem a fácil alteração do número de nós existente. No trabalho pretende-se desenvolver uma capacidade de escalabilidade muito superior à existente hoje em dia, sem processos morosos e complexos para incorporação de novos nós.
- **Tipologia multisensor** - A maioria dos Sistemas de Vigilância existentes são baseados em câmaras com captação de imagens, existindo alguns com capacidade de processar um conjunto limitado de sensores adicionais, como detetores de fumo, por exemplo. Com a arquitetura proposta, é criado um nível maior de abstração do sensor, orientando-o para detetar eventos à custa de um ou mais sensores transparentes para o sistema, que podem monitorar diferentes variáveis, passíveis de serem modificadas em qualquer momento.

- **Solução adaptável** - A reatividade dos Sistemas de Vigilância de hoje em dia passa por identificar potenciais ocorrências, ficando a cargo de um utilizador do sistema a responsabilidade de executar comandos para responder ao evento. Pretende-se, assim, criar mecanismos de resposta a eventos, envolvendo diversos tipos de sensores que podem colaborar entre si para confirmar potenciais ocorrências, mas também para manter sob vigilância todas as áreas vigiadas enquanto acompanham, de forma mais próxima, os eventos detetados. Assim, com a arquitetura proposta pretende-se aumentar as capacidades de adaptação automática de forma a responder a eventos sem perdas de capacidade de vigilância e sem estar totalmente dependente da intervenção humana.
- **Integração de vários tipos de sensores numa mesma plataforma hardware** - A integração de vários tipos de sensores permite obter um maior número de informação do local, sem necessidade de deslocação ao mesmo. Atualmente, as soluções existentes apenas proporcionam um tipo de informação, não alertando para situações conjugadas de risco, como por exemplo “fumo + inundação”. A criação de agentes que recolhem e processam dados provenientes de múltiplos sensores do tipo "*plug and play*" foi um dos objetivos deste trabalho.
- **Incorporação de múltiplas variáveis para identificação de eventos** - A maioria das soluções existentes apenas entra em conta com um tipo de informação sensorial para identificar eventos. Exemplo: sensores de contacto para detetar a abertura de portas, sensores de movimento para detetar intrusão de espaços, câmaras de vigilância

para detetar movimentos. A abordagem proposta prevê a integração de múltiplos algoritmos em simultâneo para tratar a informação proveniente de vários tipos de sensores. Esta solução permite lidar com imprecisão e incerteza tanto na combinação das variáveis, como no disparo das regras (implicação) e ainda na agregação das mesmas para chegar a uma conclusão. Exemplo: a utilização conjugada de informação proveniente de um sensor de abertura de porta com imagens de uma câmara de vigilância para detetar se uma porta se abriu devido ao vento ou por ação de uma pessoa (detetada como tal).

Estas características resumem alguns dos aspetos mais importantes na análise do estado atual das arquiteturas usadas nos sistemas de vigilância, permitindo a validação da arquitetura proposta como uma solução inovadora que permitirá melhorar a eficácia dos atuais sistemas de vigilância.

<b>Características Inovadoras</b>	<b>Situação no mercado</b>	<b>Estado da arte</b>	<b>Arquitetura proposta</b>
<b>Escalabilidade do sistema</b>	Sistemas de Vigilância com número limitado de câmaras e/ou sensores de infravermelhos	Tipicamente, nos Sistemas de Vigilância existe um número estático de sensores, quer seja câmaras quer seja sensores	Utilizando a tecnologia de agentes e funcionando cada elemento autonomamente, o número de elementos não é estático, podendo-se assim incluir ou remover agentes do sistema, de forma dinâmica, sem necessidade de o alterar
<b>Tipologia Multisensor</b>	Os Sistemas de Vigilância atuais são normalmente caracterizados por câmaras, isto é, limitados a uma tipologia de sensores	Sistemas de Vigilância com câmaras e alguma integração com alguns tipos de sensores	Definindo o agente e modo de atuação, é criada uma abstração capaz de dotar o sistema de adaptabilidade em termos de tipologia de sensores
<b>Solução Adaptável</b>	Conjunto fixo de câmaras sem inteligência para reagir a eventos	Sistemas estáticos, sem dinamismo	Utilizando cooperação entre agentes, o sistema permite a adaptação na ocorrência de um evento específico.
<b>Integração de vários tipos de sensores numa mesma plataforma hardware</b>	Câmaras de vigilância com deteção de movimento, ou sensores de infravermelhos que detetam a existência de movimento, mas não produzem mais informação.	Câmaras de vigilância com deteção de movimento, ou sensores de infravermelhos que detetam a existência de movimento, mas não produzem mais informação.	A utilização de agentes permite a introdução de qualquer tipo de sensor na plataforma, visto que, estes agentes são responsáveis por criar uma forma de comunicação entre o hardware o sistema.
<b>Incorporação de múltiplas variáveis para identificação de eventos.</b>	Sistemas atuais de vigilância utilizam algoritmos especializados para cada tipo de sensor	Processamento especializado de cada tipo de informação sensorial	A deteção de eventos é baseada em formulas dinâmicas onde é possível introduzir múltiplas variáveis.

**Tabela 3.1 - Características a analisar**

### 3.4 *Principais contribuições e Publicações*

Do ponto de vista científico foram feitas publicações dos resultados científicos em conferências científicas, tendo em conta a lista de conferências disponibilizada pelo Departamento de Engenharia e Eletrónica da FCT e o ISI WoS. As publicações feitas no âmbito deste trabalho foram:

- S Onofre, J Gomes, JP Pimentão, P Sousa, "**Data fusion of georeferenced events for detection of hazardous areas**", in 8th Advanced Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS'17), Costa de Caparica, Portugal, May 03-05, 2017, Proceedings; DOI: 10.1007/978-3-319-56077-9\_7 [76];
- Tiago Duarte, Sérgio Onofre, João Paulo Pimentão, Pedro Sousa, "**Biometric Access Control Systems: a review on technologies to improve their efficiency**", in 17th International Power Electronics and Motion Control Conference and Exposition, PEMC 2016; DOI: 10.1109/EPEPEMC.2016.7752095 [77];
- Pedro Silvestre, João Pimentão, Sérgio Onofre, Pedro Sousa, "**Surpassing Bluetooth Low Energy Limitations on Distance Determination**", in 17th International Power Electronics and Motion Control Conference and Exposition, PEMC 2016; DOI: 10.1109/EPEPEMC.2016.7752104 [78];
- S Onofre, B Caseiro, JP Pimentão, P Sousa, "**Using Fuzzy Logic to Improve BLE Indoor Positioning System**", in 7th IFIP WG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2016, Costa de Caparica, Portugal, April 11-13, 2016, Proceedings; DOI: 10.1007/978-3-319-31165-4\_18 [79];

- S. Onofre, P. Sousa, and J. P. Pimentao, “**Multi-sensor geo-referenced surveillance system**”, in 2015 10th International Symposium on Mechatronics and its Applications (ISMA), 2015, pp. 1–6 [80]; DOI: 10.1109/ISMA.2015.7373497 [80];
- S. Onofre, P. Sousa, and J. P. Pimentao, “**Georeferenced dynamic event handling**”, 6th Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS'15, in proceedings [6]; DOI: 10.1007/978-3-319-16766-4\_27 [6];
- S. Onofre, P. Sousa, and J. P. Pimentao, “**Geo-referenced multi-agent architecture for surveillance**”, Power Electronics and Motion Control Conference and Exposition (PEMC), 2014 16th International. pp. 455–460, 2014 [81]; DOI: 10.1109/EPEPEMC.2014.6980534 [81];
- J. Claro, B. Dias, B. Rodrigues, J. P. Pimentao, P. Sousa, and S. Onofre, “**Autonomous robot integration in surveillance system: Architecture and communication protocol for systems cooperation**”, Power Electronics and Motion Control Conference and Exposition (PEMC), 2014 16th International. pp. 713–719, 2014 [82]; DOI: 10.1109/EPEPEMC.2014.6980581 [82];
- B. Dias, B. Rodrigues, J. Claro, J. P. Pimentão, P. Sousa, and S. Onofre, “**Architecture and message protocol proposal for robot’s integration in multi-agent surveillance system**”, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8536 LNAI, pp. 366–373 [83]; DOI: 10.1007/978-3-319-08644-6\_38 [83];

Foram também feitas algumas publicações na plataforma de acesso livre ResearchGate:

- Sérgio Onofre, Tiago Ferreira, João Paulo Pimentão, Pedro Sousa, **"DVA - Advanced Surveillance System based on Agents"**, ResearchGate, 2013;
- Sérgio Onofre, Tiago Ferreira, João Paulo Pimentão, Pedro Sousa, **"DVA - Features"**, ResearchGate, 2013;
- Pedro Sousa, Pedro Gomes, Sérgio Onofre, Tiago Ferreira, J. Pamies Teixeira, Bruno Belo, João Paulo Pimentão, **"HOLOS - ServRobot-006 - Mobile Robot Reconfigurability"**, ResearchGate, 2013;
- Pedro Sousa, Pedro Gomes, Tiago Ferreira, João Paulo Pimentão, Hélder Silva, Sérgio Onofre, **"Determinação de Fiabilidade de Robots móveis"**, ResearchGate, 2013;

### 3.5 *Integração com outras atividades de investigação*

O trabalho desenvolvido no âmbito deste doutoramento foi, como referido, inserido no trabalho desenvolvido no seio do projeto DVA.

Houve também uma forte ligação com um segundo projeto QREN, o SERVROBOT, Robot de serviços reconfigurável, todo-o-terreno com comportamentos autónomos. Neste projeto foi desenvolvido um robot móvel de controlo autónomo e mecanicamente reconfigurável, e energeticamente eficiente, que pode ser usado em sistemas de vigilância, dotado de mobilidade e permitindo a recolha de informação sensorial do local, podendo ainda ser usado para executar determinadas tarefas em locais ambientalmente agressivos para o ser humano. A relação com este projeto foi a integração deste robot como um agente móvel, quer para deteção de eventos quer para a sua validação.

O trabalho executado no âmbito destes projetos e a investigação desenvolvida permitiu sustentar e comprovar o trabalho de doutoramento desenvolvido.

Paralelamente, e tendo por base o trabalho científico e o protótipo funcional desenvolvidos no âmbito desta dissertação, foi suportado o desenvolvimento de oito dissertações de mestrado, das quais seis já foram aprovadas, e as últimas duas serão ainda discutidas no ano letivo 2016/2017.



## **Arquitetura baseada em Multiagentes georreferenciados para Sistemas de Vigilância**

Tendo em conta o estudo efetuado sobre as várias abordagens seguidas pela comunidade científica, e as lacunas encontradas nos atuais sistemas de vigilância, neste trabalho é proposta uma nova abordagem para arquiteturas de sistemas de vigilância. Esta nova abordagem é baseada numa arquitetura multiagentes distribuída e introduzindo o conceito de posição geográfica dos seus agentes.

Esta arquitetura pretende responder a diferentes tipos de cenários, sendo dinâmica e adaptativa.

Neste capítulo, é feita a descrição pormenorizada da arquitetura e dos seus componentes. É também descrito o modelo de dados e funcionamento de um sistema baseado nesta arquitetura.

## 4.1 *Arquitetura*

Como referido, a arquitetura proposta é baseada em multiagentes, o que permite uma abordagem ao problema repartida em diversos agentes, sendo cada um deles responsável por tarefas bem específicas e que o caracterizam.

Na Figura 4.1 é ilustrada a arquitetura proposta, que para além de apresentar todos os agentes que intervêm no sistema, mostra todas as possíveis comunicações entre eles (as mensagens de configurações não foram incluídas no esquema para manter a legibilidade do mesmo, mas serão detalhadas no próximo subcapítulo). Na figura está representado o agrupamento dos agentes por tipo, sendo que cada agente possui características diferenciadoras. Os tipos de agentes existentes são: **Sensores, Processadores, Inferência, Ação, Interface, Mobile, Backup e Monitorização**, cujas funcionalidades serão detalhadas no próximo subcapítulo.

A criação de cada agente é feita de forma dinâmica permitindo a otimização de recursos computacionais e redimensionamento do sistema segundo as necessidades em cada momento. Por exemplo, à medida que os sensores se vão ligando à rede e mediante o seu tipo, serão criados sensores de processamento, fazendo com que a capacidade do sistema cresça à medida que a componente sensorial cresce.

O funcionamento da arquitetura é baseado na posição geográfica dos seus elementos, sobretudo dos Agentes Mobile e dos parâmetros produzidos pelos Agentes Processador, os quais refletem a localização onde os dados foram recolhidos pelos sensores. Serão estas posições geográficas que permitirão ao sistema responder a eventos de uma forma dinâmica e diferenciada, permitindo não só a resposta mais rápida ao evento, mas também a otimização de recursos.

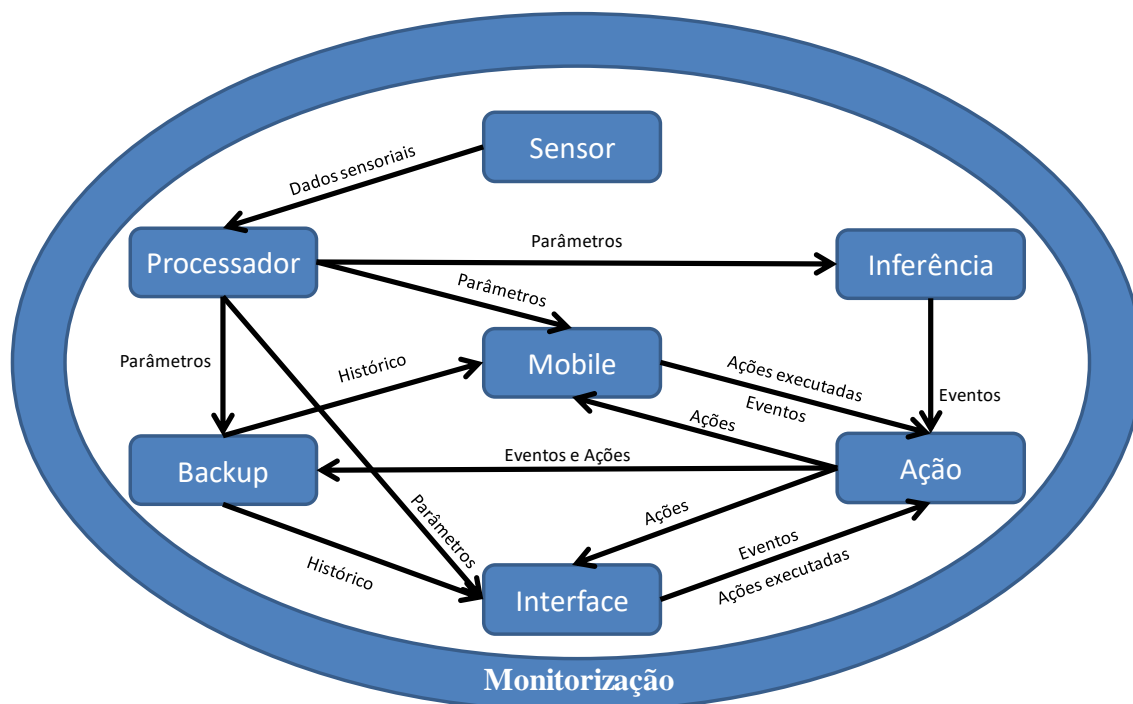


Figura 4.1 – Arquitetura do sistema

Como é ilustrado nesta figura, a abordagem ao problema é repartida por diversos agentes, sendo cada um deles responsável por tarefas específicas que o caracterizam.

A introdução de um agente **Sensor**, que ocorre por deteção de um novo sensor ligado ao sistema numa perspetiva *Plug-and-play*, implica o surgimento de um ou mais agentes **Processador** responsáveis por sua vez por extrair parâmetros a partir da informação sensorial disponibilizada. Estes agentes produzem informação que passa a ser utilizável pelo sistema de **Inferência** e é simultaneamente guardado (para prova e concretização de histórico) no agente **Backup**. Os resultados da Inferência podem ativar agentes **Ação** que se encarregam de seguir o evento gerado, notificando a **Interface** – web - ou o agente **Móvel** mais adequado, as quais são guardadas pelo agente Backup para evidência. Através da Interface é possível consultar informação histórica de parâmetros e eventos. Os agentes **Monitorização** são responsáveis por monitorar o funcionamento de todo o sistema, incluindo recuperação de falhas e identificação de novos sensores.

A criação dos agentes é feita de forma dinâmica, isto é, só à medida que os sensores se vão conectando à rede e mediante o seu tipo, os agentes são criados para realizar o seu processamento. Assim sendo, o sistema escalará em proporção direta com o crescimento da sua componente sensorial.

Um exemplo de funcionamento do sistema baseado nesta arquitetura pode ser descrito sucintamente como:



Figura 4.2 - Cadeia de processamento de informação

## 4.2 Componentes da arquitetura

A arquitetura tem como objetivo a disponibilização de um sistema escalável e facilmente adaptável “à entrada e saída” de sensores e à ocorrência de eventos, adaptando a sua tipologia e distribuição aos diferentes tipos de sensores existentes (e consequentemente parâmetros mensuráveis) e aos diferentes tipos de eventos a monitorizar.

Nos parágrafos seguintes apresenta-se a caracterização detalhada dos vários tipos de agentes que foi sumariamente descrita acima.

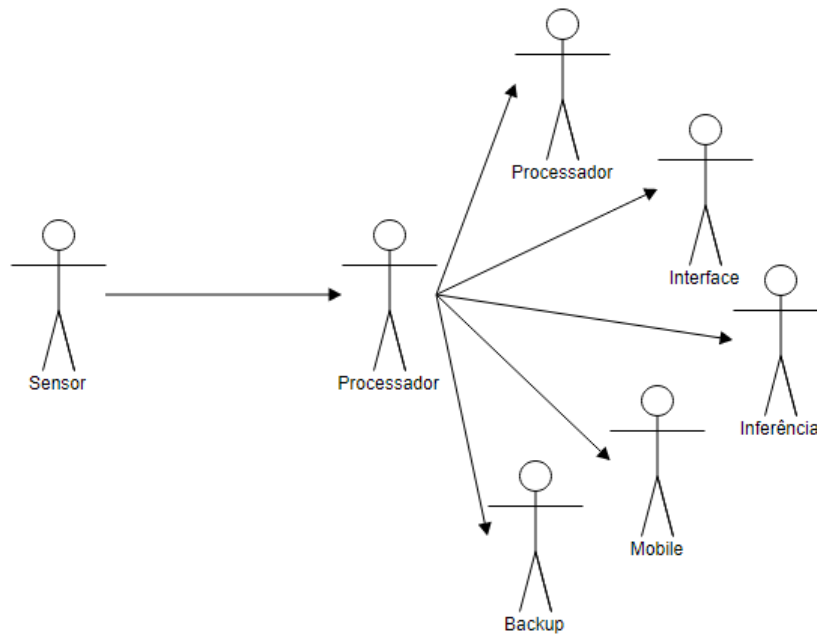
- **Agentes Sensor** - Um Agente do tipo Sensor é um agente simples, que apenas adquire sinais e valores e os disponibiliza para tratamento. Uma câmara ou um sensor de movimento são exemplos de agentes sensores, produzindo, respetivamente, imagens e um sinal que indica a presença de movimento no seu raio de deteção. São estes os agentes responsáveis por realizar a interface com o hardware específico e que integram, por exemplo, métodos de inicialização

e/ou calibração quando requerido. O Agente sensor envia a informação para o Agente Processador.



Figura 4.3 – Fluxo de informação do Agente Sensor

- **Agentes Processador** - A informação produzida pelos Agentes Sensor é passada, no seu estado bruto, para o Agente Processador que trata a informação recolhida, produzindo para o nível seguinte uma nova informação, à qual é dado o nome de parâmetro (que poderá ou não conter a informação original). Os Agentes Processador podem também produzir valores dependentes da integração/fusão de informação proveniente de vários sensores, por exemplo, juntar informação de temperatura, pressão e humidade para produzir um novo parâmetro: “probabilidade de chuva”. Neste caso, o input para o Processador não é um sensor físico, mas um conjunto de parâmetros previamente produzidos por outros agentes Processador. O Agente Processador recebe informação do Agente Sensor e envia a informação para os Agentes: Interface, Mobile, Inferência e Backup.



**Figura 4.4 – Fluxo de informação do Agente Processador**

- Agentes Inferência** – Este tipo de agente é responsável pelo processamento das regras de inferência programadas para o sistema e utiliza mecanismos de inferência para a geração de eventos, cruzando a informação proveniente dos vários sensores (devidamente processada por agentes do tipo Processador) e tendo em conta o seu raio de deteção (área que o sensor consegue monitorizar; ex.: área de uma sala onde está instalado um sensor de temperatura). A inferência é realizada constantemente com os parâmetros enviados pelos Agentes Processador, sendo que os eventos gerados dão origem à criação de um Agente do tipo Ação para o seu tratamento. Entre outras, o sistema de inferência está preparado para lidar com regras temporais que permitem uma análise mais fidedigna das situações e mais imune a ruído (e.g. “se  $A > 56$  e  $B < 33$  durante 10 segundos então  $C$ ”). O Agente Inferência recebe informação dos Agentes Processador e envia a informação para o Agente Ação.

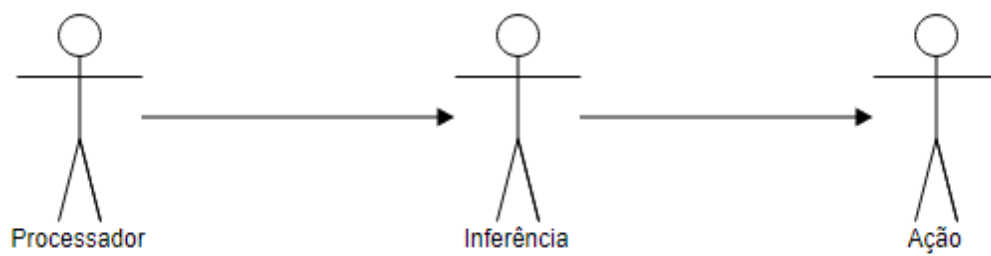


Figura 4.5 – Fluxo de informação do Agente Inferência

- **Agentes Ação** – Os Agentes deste tipo são responsáveis pelo tratamento e acompanhamento dos eventos, executando e coordenando todas as ações associadas a determinado evento, como por exemplo atribuição, envio de SMS ou emails. Para cada evento será criado um Agente do tipo Ação, sendo que este agente só existirá durante o tempo que o evento está ativo, executando todas as ações previstas. Estes Agentes recebem a informação do Agente Inferência e enviam/recebem informação para o Agente Mobile e Interface. Também envia informação para o Agente Backup.

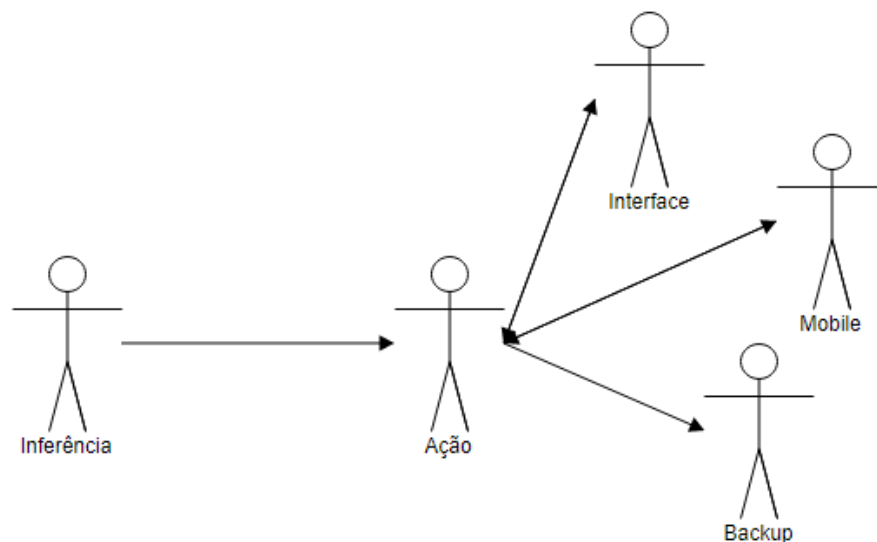


Figura 4.6 – Fluxo de informação do Agente Ação

- **Agentes Mobile** – O Agente do tipo Mobile é um dos responsáveis pela comunicação entre a plataforma de agentes e os utilizadores, permitindo, por exemplo, alertar o humano para a ocorrência de eventos, submeter eventos criados pelo humano ou consultar o estado atual do sistema. Este agente terá também associadas coordenadas geográficas, sendo que a informação recebida será enviada por um Agente Ação que leva em conta a sua localização e perfil para a afetação de ações a elementos da equipa de segurança. O Agente Mobile recebe informação dos Agentes Processador e Backup e comunica com o Agente Ação.

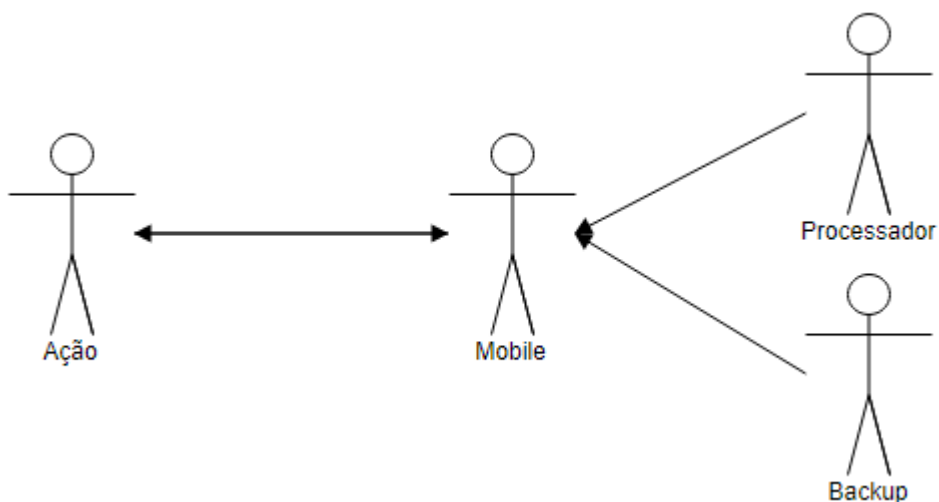


Figura 4.7 – Fluxo de informação do Agente Mobile

- **Agentes Interface** - este tipo de agente é responsável pela apresentação dos dados do sistema (*real-time* ou históricos), interagindo diretamente com os processadores (para dados em *real-time*) ou com o Agente Backup (para dados históricos). Tal como o Agente Mobile, este Agente também permite a interação como o Agente Ação, sendo que a principal diferença é que este agente tem sempre as coordenadas fixas. O Agente Interface recebe informação dos Agentes



Processador e Backup e comunica com o Agente Ação. Este agente também é responsável por enviar as configurações para os Agentes Processador, Inferência e guardá-las através do Agente Backup.

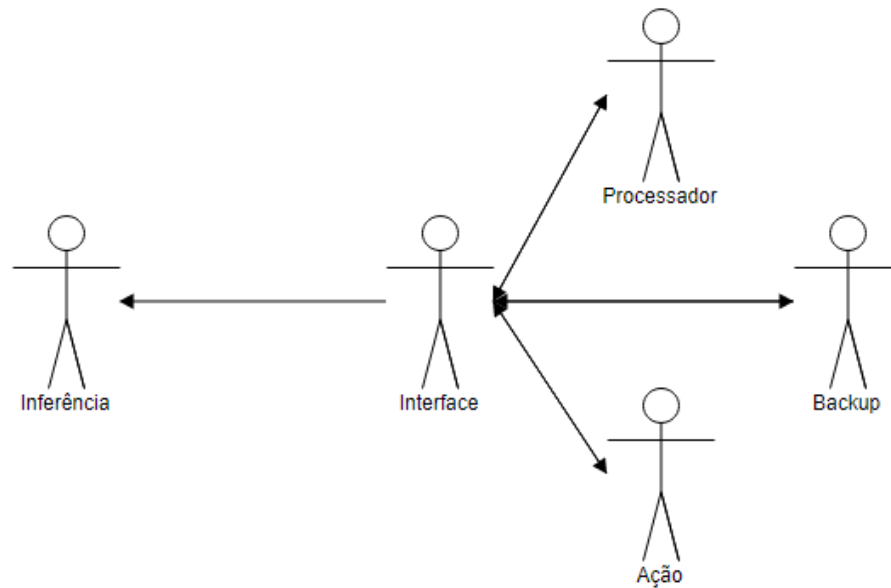


Figura 4.8 – Fluxo de informação do Agente Interface

- **Agentes Backup** – Tipo de agente responsável pelo armazenamento e disponibilização de dados históricos produzidos pelos diferentes elementos do sistema. Este agente recebe a informação de todos os acontecimentos do sistema, dados dos vários sensores, configurações, regras, etc., disponibilizando essa informação a todos os agentes que a requeiram. O Agente Backup recebe informação dos Agentes Processador e Ação e envia para os Agentes Mobile e Interface.

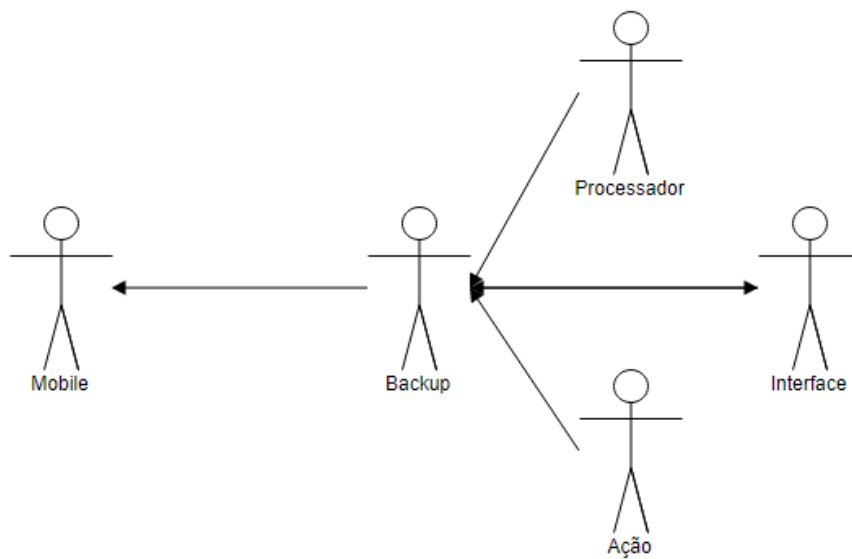


Figura 4.9 – Fluxo de informação do Agente Backup

- **Agente Monitorização** – Tipo de agente responsável pela criação inicial do Agente Inferência e Agente Processador de parâmetros calculados. Este agente também será responsável pela monitorização e integridade do sistema, verificando ciclicamente a existência de todos os agentes essenciais para o sistema assim como a carga associada a cada um dos pontos de computação. Em caso de desaparecimento de algum agente, o Agente de Monitorização tenta recriar o agente em falta ou/e cria um evento de falha.

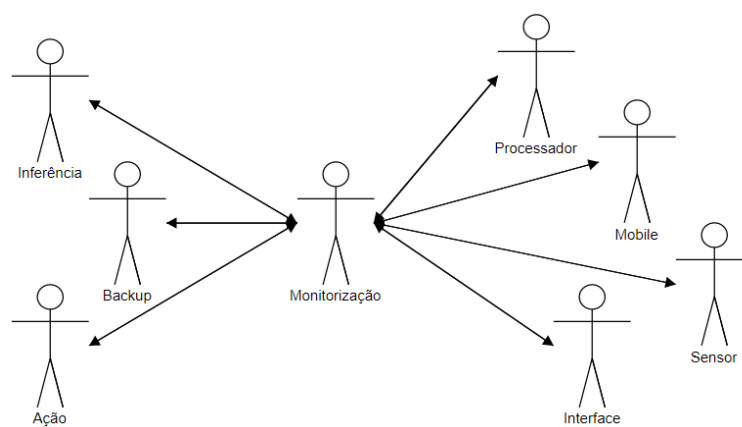


Figura 4.10 – Fluxo de informação do Agente Monitorização

## 4.3 Modelo de dados

### 4.3.1 Parâmetros

Cada sensor existente no sistema poderá produzir um ou mais dados para o sistema, como por exemplo, uma temperatura ou distância. Paralelamente poderão ser definidos novos valores que resultem da combinação de valores entre sensores ou da transformação desses valores.

Estes dados, produzidos diretamente pelos sensores ou calculados, são designados como Parâmetros e caracterizados por um nome, valor, unidade, tipo (Temperatura, Distância, Luminosidade), data, localização e origem.

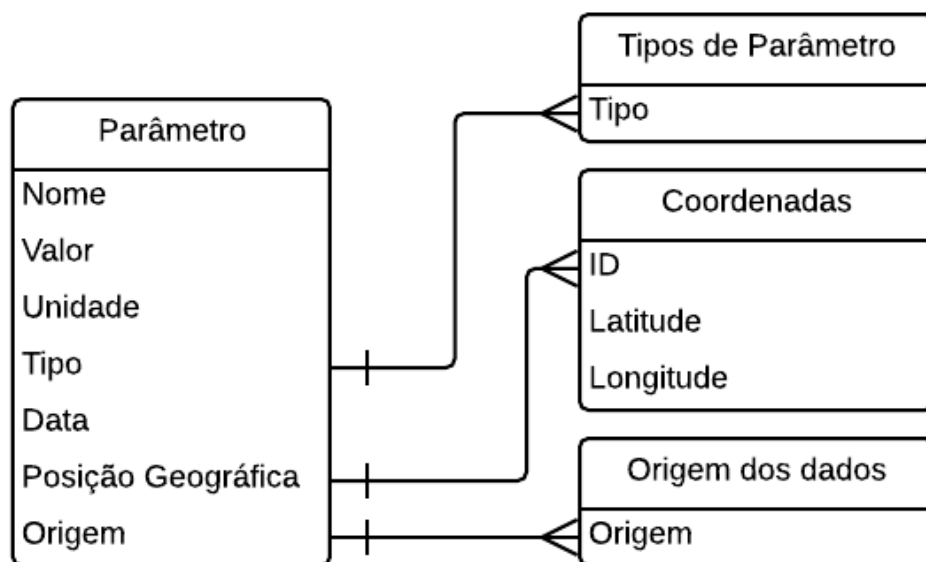


Figura 4.11 – Parâmetros

### 4.3.2 Regras

Com base nos parâmetros existentes no sistema são definidas regras para detecção de eventos. As regras representam condições definidas pelos utilizadores segundo os valores dos parâmetros, por exemplo, “temperatura maior que 50°C”. As regras definidas são a base para o motor de inferência, que constantemente avalia os valores dos parâmetros e dispara eventos caso as condições determinadas se verifiquem.

Neste sistema podem ser definidas regras que contenham:

- Condições de comparação (maior, menor, igual...) entre parâmetros ou entre parâmetros e valores;
- Condições lógicas (AND, OR, NOT...);
- Condições Temporais (durante “X” tempo);
- Regras (usar o resultado de outra regra como parâmetro).

Um exemplo de uma regra simples (formato legível):

*IF (Parâmetro1 > 30) THEN EVENTO\_INCENDIO*

Um exemplo de uma regra temporal (formato legível):

*IF (Parâmetro1 > 20 AND Parâmetro2 = 5) (DURING 120 SECONDS) THEN  
EVENTO\_INCENDIO*

### 4.3.3 Eventos

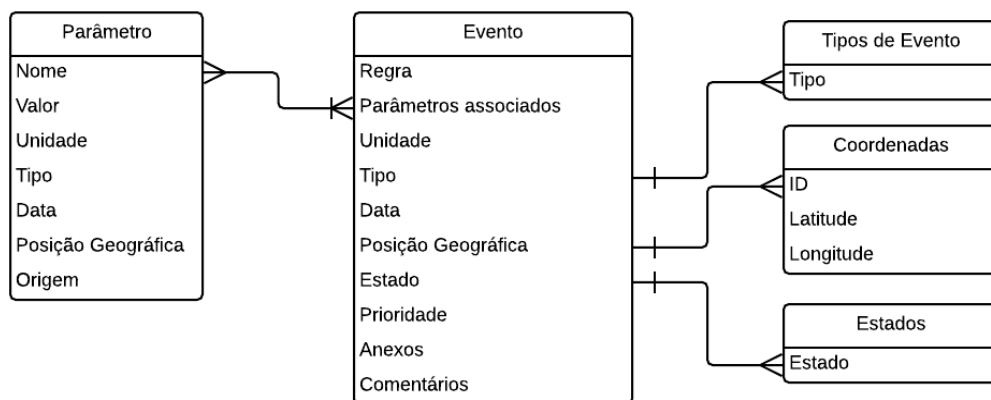
Os eventos são os acontecimentos detetados pelo sistema, isto é, segundo as regras definidas, o sistema irá alertar para a existência de um Incêndio, uma Inundação, uma Intrusão, etc. Os eventos são caracterizados por:

- Data – data e hora da ocorrência do evento;
- Posição geográfica – coordenadas da posição central do evento (caso seja só um sensor, será as suas coordenadas, caso seja com origem em vários sensores, serão calculadas as coordenadas centrais<sup>8</sup>);
- Regra – identificação da regra que lhe deu origem;

---

<sup>8</sup> As coordenadas centrais podem ser calculadas utilizando os serviços da Google ou caso não seja possível aceder aos serviços, utilizando a fórmula para o cálculo de centro de massa (com as massas igual a 1).

- Parâmetros associados – valores associados à regra que disparou;
- Tipo de evento – Incêndio, Inundação, Intrusão, Terramoto, Gás e Falha de agentes. Poderão também ser definidos novos tipos;
- Estado – estado atual do evento (estados descritos mais abaixo);
- Prioridade – Alta, média ou baixa;
- Comentários – comentários adicionados durante o tratamento do evento;
- Anexos – Imagens, sons ou vídeos associados ao evento (evidências).



**Figura 4.12 – Eventos**

Cada evento passará por diversos estados, podendo existir ações associadas a cada um deles. Os estados em que cada evento poderá estar são:

- **Por confirmar** – Estado inicial do evento, que aguarda a confirmação de um utilizador presencialmente ou por análise da informação que deu origem ao evento (em dispositivos móveis ou interface web). Esta confirmação pode ser positiva ("verdadeiro") ou negativa ("falso");

- **Confirmado** – Depois do utilizador confirmar a existência do evento, este fica em estado “confirmado” até todas as ações terem sido concluídas;
- **Concluído** – Estado final do evento, depois de terem sido executadas todas as ações predefinidas e o evento ser dado como “concluído”;
- **Cancelado** – Estado final do evento para os casos em que um utilizador verificou que não existia causa para evento.

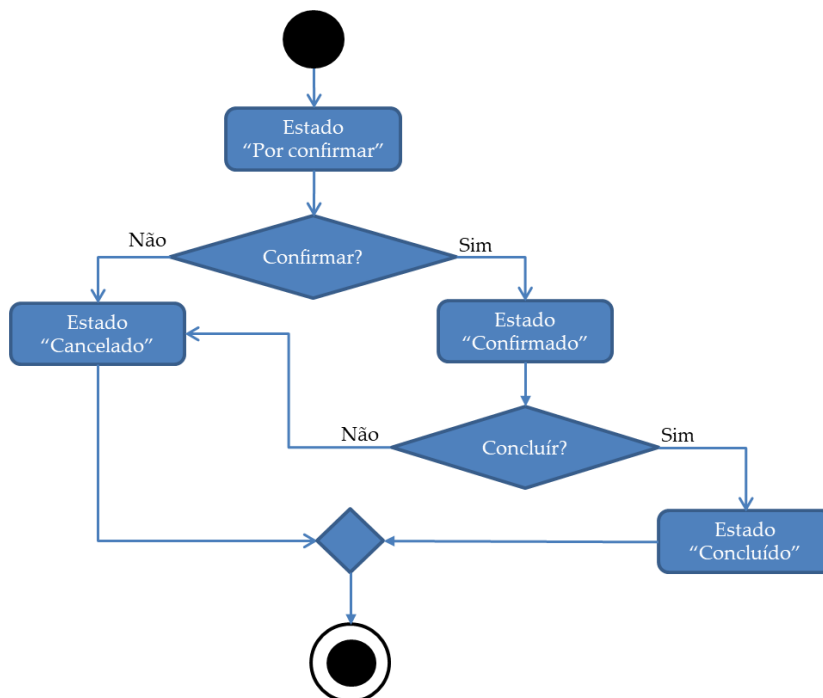


Figura 4.13 – Diagrama de estados dos eventos

A evolução dos eventos pelos vários estados está ilustrada na Figura 4.13, sendo que o estado inicial do evento será o estado "Por confirmar" e os estados finais poderão ser Cancelado ou Concluído. A transição entre estados será feita pelo utilizador que mediante a informação apresentada pelo sistema decidirá a ação a executar (Confirmar, Cancelar ou Concluir).

#### 4.3.4 Ações

Como referido na descrição dos eventos, cada estado do evento poderá ter ações associadas. As ações são tarefas específicas que o sistema executará num dado momento tendo em conta o definido pelos utilizadores. As ações disponíveis para cada estado são:

- Estado **“Concluído”** e **“Cancelado”**:
  - Envio de email de notificação;
  - Envio de sms;
- Estado **“Por confirmar”** e **“Confirmado”**:
  - Envio de email de notificação;
  - Envio de sms;
  - Chamada para serviços de urgência (apenas no estado **“Confirmado”**);
  - Atribuir evento para confirmação (**“Por confirmar”**);
  - Atribuir evento para tratamento (**“Confirmado”**).

Para cada uma das ações definidas é também possível definir uma ação de recurso dentro das opções já enumeradas, que será executada caso o sistema não consiga realizar a ação definida. Por sua vez, cada ação de recurso poderá ter outra ação de recurso. Isto é, poderá existir uma hierarquia de ações, permitindo para cada ação definir uma ação de recurso, sendo que estas só serão executadas no caso de a ação superior falhar. Por exemplo, se a atribuição a um Agente Mobile falhar, por não estar nenhum disponível, pode definir-se o envio de um sms ao administrador para tratar do problema. Desta forma, o sistema permite dar resposta a eventuais problemas associados a ações previstas, mas não executadas.

Durante a fase de tratamento do evento também é possível:

- Adicionar detalhe ao evento (fotos, vídeos, texto);
- Adicionar outro agente do tipo Ação ao evento;
- Reatribuição do evento caso o agente responsável se tenha desligado do sistema, ou seja, desafetado do evento;
- Desafetação de um evento;
- No caso específico de uma Intrusão: Seguimento do suspeito.

#### 4.4 *Funcionamento do sistema*

Cada elemento desta arquitetura tem um funcionamento específico e interligado com o funcionamento de outros elementos. Na base do funcionamento do sistema estão os sensores que monitorizam parâmetros físicos, como por exemplo, a temperatura. Cada sensor adicionado ao sistema dará origem a um **Agente Sensor**, responsável pela comunicação entre o sensor (hardware) e o sistema (software). É o Agente Sensor o responsável pela realização de operações de calibração e pela aquisição de dados.

Cada Agente Sensor dará, por sua vez, origem a um **Agente Processador**, que transformará a informação sensorial e disponibilizará ao resto do sistema essa informação sobre a forma de parâmetro (ou conjunto de parâmetros, dependendo do sensor) com uma coordenada e raio de detecção associados. O sistema também permite definir áreas, isto é, delimitar um determinado espaço e atribuir um nome a esse espaço, permitindo assim enquadrar as coordenadas nesse espaço. Desta forma a informação sensorial pode ser apresentada como “temperatura em graus Celsius na sala 34 do edifício III”.

Paralelamente, os utilizadores (administradores de sistema) poderão criar Agentes Processadores adicionais, quer para a disponibilização da informação obtida do sensor sobre outra forma como por exemplo, “Temperatura em graus



Fahrenheit”, quer para o cálculo de novos parâmetros (por exemplo, a temperatura média obtida de dois sensores).

A criação dos **Agentes Processador** que fornecem parâmetros calculados é feita pelo **Agente Monitorização** que, no arranque do sistema e após consultar o **Agente Backup** (que fornece a informação de quais os Agentes Processador a criar tendo em conta a meta-informação guardada), cria os Agentes Processador necessários, segundo as fórmulas existentes. Durante a execução normal do sistema, se for criado um novo parâmetro, será o **Agente Monitorização** a criar automaticamente o Agente Processador que permitirá disponibilizar o novo parâmetro ao sistema.

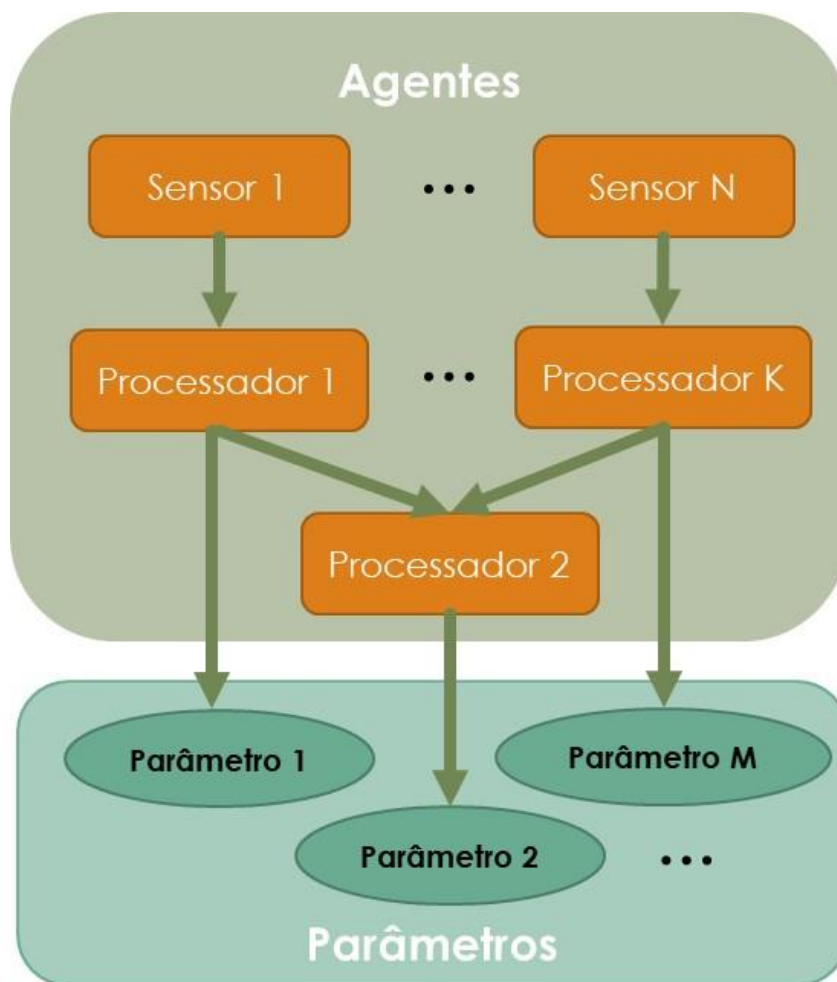


Figura 4.14 – Geração de Parâmetros

Estes parâmetros, para além de consultáveis pelos utilizadores, são utilizados nas regras de inferência definidas para o sistema. Estas regras são definidas pelos utilizadores e executadas pelo **Agente Inferência**, tendo em conta os parâmetros existentes, a localização dos sensores e outros critérios que se possam considerar (exemplo: a hora do dia). Cada regra avaliada com sucesso dará origem a um evento, categorizado e geolocalizado. O sistema poderá ter mais do que um Agente Inferência, ficando cada um responsável por uma determinada área.

O Agente Inferência será o responsável por criar os **Agentes Ação** para eventos, segundo o tipo de evento detetado. No arranque do sistema, por paragem planeada ou inesperada, será o Agente Monitorização o responsável pela criação do Agente Inferência.

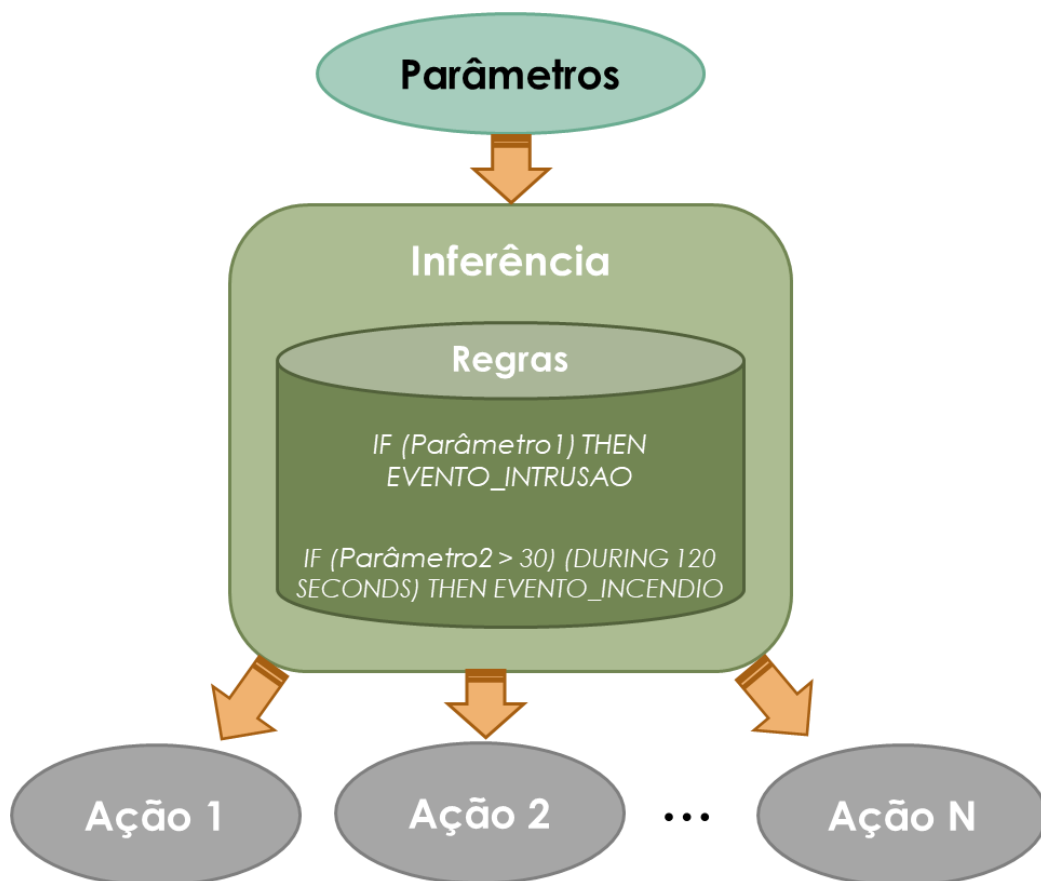


Figura 4.15 – Geração de Ações

Um Agente Ação é criado para cada evento e terá a duração de vida do evento, isto é, é criado no aparecimento do evento e eliminado quando o evento estiver concluído. Cada Agente Ação é responsável pela execução e coordenação de todas ações definidas pelo utilizador para um dado tipo de evento e área. Para cada área, e tendo em conta o tipo de evento, pode ser definido um conjunto de ações, por exemplo:

- caso tenha sido detetada uma Inundação na “sala de servidores”, o Agente Ação enviará um sms de alerta ao segurança responsável
- caso exista um incêndio na “sala de servidores” o Agente Ação poderá contactar automaticamente os bombeiros mais próximos do local ou tomar ações semelhantes de alerta.

Para além da parametrização por área e tipo de evento também é possível diferenciar ações por estado do evento. Por exemplo, para o estado “Por Confirmar” pode ser definido que deve ser notificado um **Agente Mobile** mais próximo do acontecimento para que avalie o evento e que caso o evento seja “confirmado”, ligar automaticamente para os bombeiros.

O Agente Ação também pode fazer a relação entre eventos, criando uma ligação entre os eventos no detalhe de cada evento e notificando os agentes desta relação. Esta ligação entre eventos é feita através da correlação das coordenadas geográficas, data/hora da ocorrência, tipo de evento e detalhes do evento (por exemplo, imagem do intruso). Por exemplo, se ocorrerem dois eventos do tipo Intrusão num espaço de 5 minutos em coordenadas adjacentes (por exemplo 50 metros distantes) o sistema alerta para a hipótese de estes eventos estarem correlacionados. No entanto, se a distância entre os eventos for de 5km e a diferença entre a hora de ocorrência for de 30 segundos, o sistema não alerta para uma possível correlação.

O Agente Mobile, tal como descrito anteriormente, poderá ser notificado de um evento, sendo que esta notificação poderá ser feita em diferentes momentos. Cada Agente Mobile terá um utilizador associado, podendo o mesmo,

para além de responder a solicitações do sistema, executar os seguintes comandos:

- Criar no sistema novos eventos por si visualizados;
- Consultar alguns dados do sistema, como por exemplo: valores e localização de parâmetros;
- Localizar outros Agentes Mobile no mapa;
- Consultar a lista de eventos recebidos e criados;
- Confirmar/cancelar eventos;
- Tratar eventos.

Para além do Agente Mobile, também o Agente Interface é ser usado para confirmar, tratar ou criar novos eventos. Através do Agente Interface será também possível consultar toda a informação do sistema, tanto em tempo real como informação histórica, configurar regras de inferência, ações para os eventos, definir áreas, configurar parâmetros e utilizadores. O Agente Interface está disponível através de um site web que permite aceder a todos estas funcionalidades.

Paralelamente a todo o sistema existe uma base de dados, onde são guardadas todas as informações do sistema e os dados históricos. O **Agente Backup**, para além das funções anteriormente definidas é o agente responsável por guardar informação e disponibilizá-la ao resto dos agentes. Entre outras informações, são guardadas as regras do Agente Inferência, as áreas, as ações para cada um dos tipos de Evento e área, fórmulas dos parâmetros calculados e os utilizadores e seus perfis.

Conforme referenciado anteriormente, o **Agente Monitorização**, para além da criação de Agentes Processador e Inferência no arranque do sistema, também irá periodicamente verificar o funcionamento e integridade do sistema,

detetando por exemplo a falta de um agente e recriando o mesmo. As principais tarefas deste agente são:

- No arranque do sistema:
  - Criar o Agente Backup;
  - Criar o Agente Inferência;
  - Criar os Agentes Processadores para os parâmetros calculados;
  - Detetar eventos não concluídos e criar os Agentes Ação correspondentes.
- Verificar ciclicamente se os agentes base estão a funcionar: Backup, Interface e Inferência;
- Verificar ciclicamente se os agentes gerados estão a funcionar: Processadores calculados, Ação de eventos a decorrer.

O sistema estará também a monitorizar a existência deste agente com regras especiais que, caso falhem, terão como consequência a recriação do Agente Monitorização.

Em forma de resumo, na Tabela 4.1 são descritas as mensagens trocadas entre agentes.

Destino Origem	Sensor	Processador	Inferência	Ação	Mobile	Interface	Backup
Sensor	-	Dados Sensoriais	-	-	-	-	-
Processador	-	Parâmetros	Parâmetros	-	Parâmetros	Parâmetros	Parâmetros
Inferência	-	-	-	Eventos	-	-	-
Ação	-	-	-	-	Ações	Ações	Eventos e Ações
Mobile	-	-	-	Ações Executadas e Eventos	-	-	-
Interface	-	Configurações	Configurações	Ações Executadas e Eventos	-	-	Configurações
Backup	-	-	-	-	Histórico	Histórico	-

**Tabela 4.1 - Mensagens entre agentes**

Adicionalmente, na fase final do trabalho foi feita uma integração com um projeto de investigação o ServRobot, mencionado anteriormente, que permitiram a integração de robots móveis como agentes do sistema, quer a nível de disponibilização de dados sensoriais, quer para validação *in loco* de eventos. Neste caso foram definidos dois novos tipos de agente, o Robot Sensor e o Robot Móvel.

O Robot Sensor é o agente responsável pela comunicação entre o sistema de vigilância e o robot em termos de dados sensoriais, disponibilizando os mesmos no formato idêntico ao disponibilizado pelos outros agentes sensores. O Robot móvel terá um funcionamento semelhante ao Agente Mobile, permitindo a atribuição de tarefas de confirmação de eventos[83].

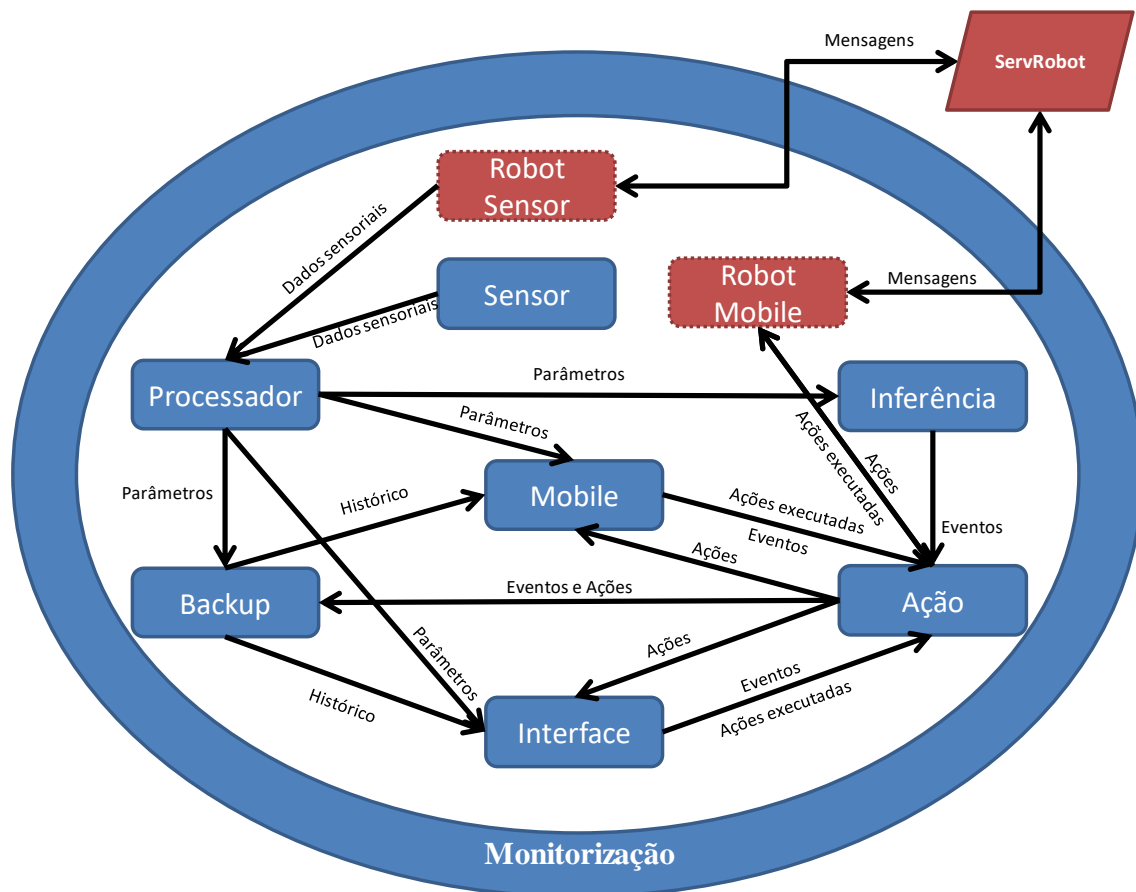


Figura 4.16 – Arquitetura estendida [83]

A comunicação entre estes dois sistemas, Sistema de Vigilância e Robot Móvel é feita através de um novo protocolo de comunicação, implementado nos dois novos agentes e pelo Robot Móvel[82].

#### **4.4.1 Atribuição georreferenciada dinâmica de um Recurso a um Evento**

Existem várias abordagens para o problema de alocação dinâmica de tarefas, e vários fatores que podem ser levados em conta na distribuição das mesmas, tais como o tipo de tarefa, localizações ou desempenho dos recursos. Assim, os parâmetros usados para a atribuição de uma tarefa podem implicar uma significativa quantidade de dados para correlacionar.

Uma das abordagens para visualizar e interpretar grandes conjuntos de dados multidimensionais é o *Self Organizing Map* (SOM). O SOM pode ser usado para visualizar estados de processos ou resultados financeiros, representando as dependências centrais com os dados sobre o mapa [84]. Em relação à atribuição de tarefas baseada em redes neurais SOM, uma das mais recentes abordagens é descrita por Huan Huang et al em [85], onde é proposta uma atribuição de tarefas dinâmica e planeamento de trajeto para sistemas de Multi-AUV (veículos submarinos autónomos) em ambientes de correntes oceânicas variáveis. Neste trabalho, os autores apresentam dois algoritmos utilizados para resolver este problema: algoritmo baseado em SOM - para atribuir metas a AUVs; síntese de velocidade - para planear o caminho para cada AUV no complexo ambiente oceânico atual.

Outra abordagem para os AUV é apresentada por Sanem Sariel et al em [86], onde a plataforma distribuída DEMiR-CF é usada para distribuir tarefas a equipas Multi-robot. Esta estrutura integra sistemas incrementais de seleção de tarefas, métodos de alocação distribuídos e várias rotinas de prevenção. Também baseado em SOM, Anmin Zhu e Simon X. Yang em [87], propõem uma atribuição dinâmica de tarefas para múltiplos robots em ambientes arbitrários não estacionários, onde são: definidos os pesos neurais iniciais da rede para seleccionar o vencedor; atualizados os pesos, e definidas funções de vizinhança.

Em [88], é apresentado um algoritmo *health-aware* de atribuição de tarefas para controlo e coordenação de múltiplas ações de veículos autônomos em ambientes dinâmicos, melhorando a fiabilidade e capacidade operacional das equipas através de um sistema de *self-awareness* e planeamento de missões adaptativas.

Um outro algoritmo de coordenação, baseado no mercado distribuído para equipas de robots, é apresentado por Nathan Michael et al em [89], onde os agentes disputam a atribuição de uma tarefa, assumindo-se que estes agentes têm conhecimento de todas as tarefas, bem como o número máximo de agentes que podem ser atribuídos a cada tarefa individual.

Brandon J. Moore e Kevin M. Passino em [90], descrevem um algoritmo em formato de leilão distribuído, usado para atribuir agentes móveis a tarefas distribuídas espacialmente. Esta abordagem controla o movimento dos agentes durante o progresso do algoritmo.

Finalmente, no artigo de James McLurkin e Daniel Yamins em [91], são descritos quatro algoritmos para atribuição dinâmica de tarefas em sistemas multi-robot: *Random-Choice*, *Extreme-Comm*, *Card-Dealer's* e *Tree-Recolor*.

Todas estas abordagens podem ser utilizadas em diferentes sistemas, tal como descrito nos artigos mencionados, no entanto, todos eles assumem que os agentes possuem as mesmas características, não diferenciando a sua adequabilidade para uma dada tarefa.

Nestas abordagens, também não está prevista a execução de diferentes tipos de tarefas, assumindo-se que todas têm o mesmo tipo de complexidade e tempo de execução. Estes pressupostos funcionam na maioria dos sistemas descritos, no entanto, em sistemas que têm diferentes tipos de tarefas e diferentes tipos de agentes para executá-las, já não podem ser aplicadas diretamente estas abordagens.

Estes algoritmos também têm em comum a suposição que são conhecidas todas as tarefas que irão ser executadas no início da atribuição de tarefas. Este



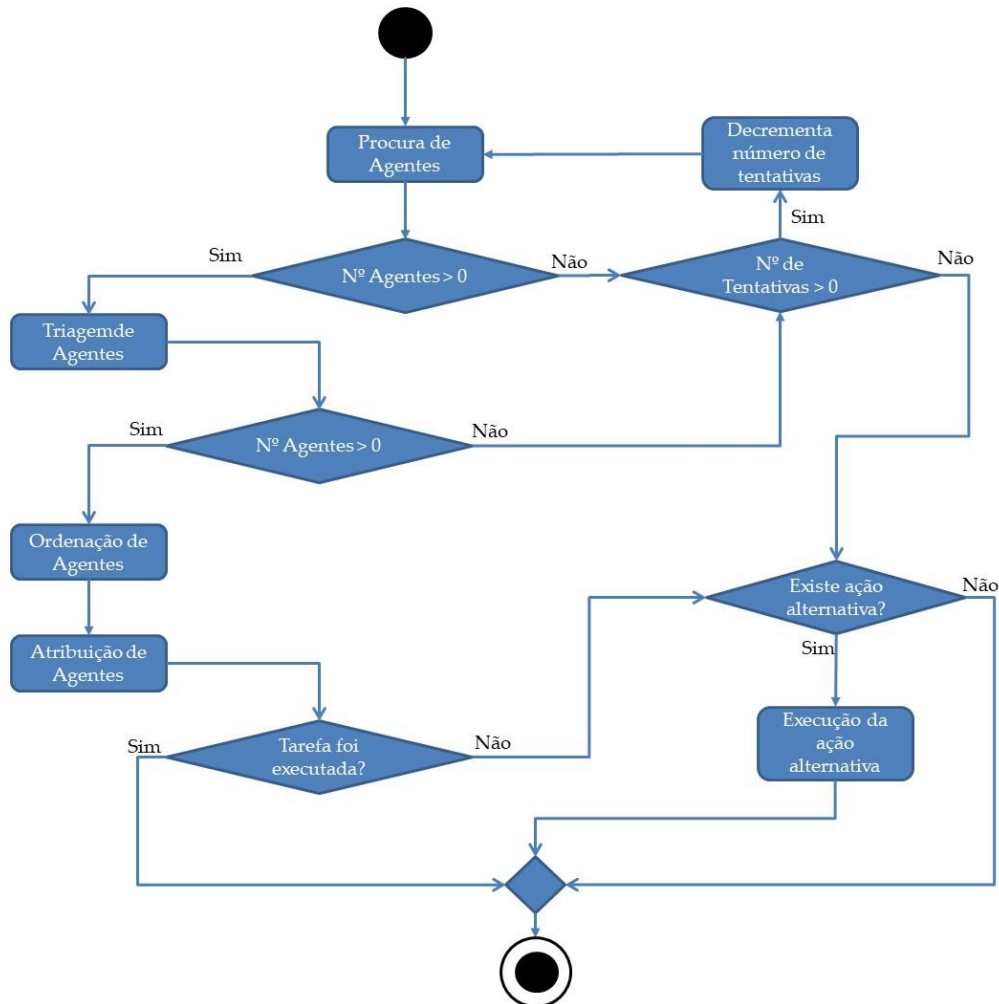
facto não pode ser assumido em sistemas de vigilância, em que os eventos podem surgir em qualquer altura e com diferentes prioridades, podendo alterar inclusive o tratamento de tarefas em curso.

Tendo as limitações das soluções encontradas para a atribuição de eventos, foi desenvolvida, também no âmbito deste trabalho e publicada em **Georeferenced dynamic event handling** [6], uma nova abordagem para a atribuição dinâmica de eventos, descrita seguidamente.

A cada evento poderá ser atribuído um ou mais recursos(Agentes), quer para validação da ocorrência, quer para o tratamento da mesma. Para a atribuição de um evento, podem ser definidos diversos critérios para filtrar e ordenar os recursos disponíveis, tais como:

- Número de recursos a atribuir;
- Tempo de resposta do Agente à proposta do Agente de Ação, isto é, aceitação da mensagem para tratamento do Evento (um exemplo de falha/longo tempo de resposta, pode ser o telemóvel estar sem rede ou desligado);
- Tempo de resposta à confirmação/tratamento do evento;
- Número de tentativas de pesquisa de Agentes;
- Número de tentativas para atribuição, em caso de falha;
- Tipo de Agente: Mobile ou Interface;
- Perfil: Bombeiros, Polícias, Seguranças e Proteção Civil;
- Área de ação do recurso (Ex.: Segurança responsável para vigiar a parte sul de um parque industrial);
- Disponibilidade: se está online ou não.

O algoritmo seguido para atribuição de eventos, ilustrado na Figura 4.17 começa por uma pesquisa no sistema de todos os agentes que cumpram com os critérios definidos.



**Figura 4.17 – Diagrama de estados de atribuição de eventos**

Depois de receber a resposta de todos ou atingido o tempo limite para resposta dos agentes, são filtrados os agentes segundo o perfil do seu recurso (caso tenha sido explicito um tipo específico de perfil) e a sua área de atuação. Tanto nesta filtragem como no caso anterior de pesquisa de agentes, se o número de agentes obtidos não for suficiente para os critérios definidos (número de agentes), é feita uma nova tentativa para encontrar agentes, aguardando o tempo definido para execução dessa nova tentativa. O número de tentativas para obter agentes será também segundo o critério definido.

Após ter sido obtido o conjunto de agentes que respeitam os critérios pré-definidos, os agentes(recursos) são classificados e ordenados segundo essa classificação. O recurso melhor classificado, tendo em conta os critérios para classificação de recursos (descrito na seção seguinte), será contactado para atribuição do evento. Caso não responda em tempo útil (tempo pré-definido) e depois de esgotadas as tentativas previstas será executada a ação de recurso, caso exista.

Depois de atribuído o recurso, o sistema aguardará a sua resposta quer no caso de confirmação/cancelamento do evento, quer no caso do tratamento do evento. No caso da confirmação/cancelamento, se o evento for confirmado, será feita uma nova pesquisa de agentes para atribuição de agentes, segundo os critérios definidos para esse estado, repetindo-se o ciclo descrito.

No processo de pesquisa de agentes, assim como na atribuição, foi usado o CFP (*Call For Proposal*) do FIPA *Contract Net Interaction Protocol*, que permite fazer a negociação entre os agentes atribuídos aos recursos e os agentes Ação. O uso deste protocolo permite simplificar eventuais problemas com a comunicação, tempos de reposta, etc.

#### **4.4.1.1 Classificação de Recursos**

Para a classificação de recursos disponíveis no momento no sistema, tendo em conta a sua atribuição a um evento, foram usadas as seguintes premissas:

Categorização de eventos por:

- tipo (incêndio, terremoto, inundação, intrusão, gás e falha de agentes);
- prioridade (alta, média, baixa);
- localização (coordenadas GPS).

Categorização de recurso por:

- perfil (bombeiros, polícias, segurança e proteção civil);
- carga de trabalho (número de eventos atribuídos);

- localização (coordenadas GPS);
- tipo de locomoção (a pé, de carro, moto, etc.);

Usando estas categorias como critério pode-se classificar e selecionar o Recurso mais adequado à atribuição do evento.

### **Tempo e distância**

Usando as novas tecnologias como os serviços do Google Maps para calcular distâncias e tempos de viagem (a pé ou de carro), é possível usar estes valores como parâmetros para a definição de qual o recurso mais perto do evento.

No entanto, para assegurar a independência de serviços externos, em caso de falha destes, pode também ser usado para o cálculo da distância, em Km, a fórmula de Haversine [92]:

$$D = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (1)$$

Onde:

- $D$  – Distância entre duas localizações (em Km);
- $r$  – Raio da Terra (6371 Km);
- $\phi_1, \phi_2$  - Latitude do local 1 e local 2;
- $\lambda_1, \lambda_2$  - Longitude do local 1 e local 2;

Usando a distância calculada é possível estimar o tempo médio para alcançar um local, de carro ou a pé, dependendo da sua velocidade média. Por exemplo, se considerarmos 5Km/h como velocidade média para andar a pé e 50Km/h para deslocamentos de carro, vamos ter:

$$T_{walking} = \frac{D}{5} \times 60 \quad \text{e} \quad T_{car} = \frac{D}{50} \times 60 \quad (2)$$

Como mencionado, o tempo mínimo para alcançar o local do evento será usado como um dos critérios para a classificação dos recursos.

### **Carga de trabalho**

Cada recurso pode ter um ou mais eventos atribuídos, representando cada evento, um aumento da sua carga de trabalho. O mesmo número de eventos pode não representar um mesmo tempo de ocupação, visto que cada evento pode ter um tempo de resolução diferente. Outro dos fatores pode ser também o tipo de evento, uma vez que cada tipo de evento pode ter associada uma complexidade de resolução diferente.

Esta indeterminação do tempo de resolução, aumenta a importância da carga de trabalho dos recursos na altura de atribuição de um evento. Para simplificação, na apresentação deste critério, e como não é possível determinar o tempo de resolução para um dado evento, o critério usado para a classificação dos recursos será apenas o número de eventos atribuídos. Na implementação final deste sistema, este critério foi baseado na análise do histórico do tempo para tratamento de eventos, tendo em conta o seu tipo.

### **Perfil**

Dependendo do tipo de perfil dos recursos, a sua capacidade de resposta a um dado evento poderá ser diferente, por exemplo, um polícia terá mais dificuldades para combater um incêndio que um bombeiro, assim como o bombeiro terá mais dificuldades para lidar com uma intrusão.

Tendo em conta as possíveis limitações impostas pelo perfil dos recursos no tratamento de um evento foi classificada a adequabilidade de um determinado tipo de perfil a um dado evento. Na Tabela 2.1, é ilustrado um exemplo para este tipo de classificação onde 1 representa o melhor desempenho e 5 o pior para a resolução de cada tipo de evento.

	<b>Polícia</b>	<b>Bombeiro</b>	<b>Segurança</b>	<b>Proteção Civil</b>
<b>Incêndio</b>	3	1	3	3
<b>Inundação</b>	3	1	4	2
<b>Intrusão</b>	1	5	2	5
<b>Terremoto</b>	3	2	4	2
<b>Gás</b>	3	3	4	2
<b>Falha de agentes</b>	5	5	3	5

**Tabela 4.2 – Perfil versus performance**

### **Prioridade e lista de eventos**

Quando é desencadeado um novo evento, os recursos podem estar livres, a caminho de um evento, ou a tratar um evento. Se o recurso não estiver livre, a prioridade do novo evento deverá ser confrontada com a prioridade do evento atual ou lista de eventos atribuídos, podendo alterar a prioridade do tratamento do recurso.

Tendo em conta as prioridades, a lista de eventos pendentes de cada recurso, poderá também influenciar a distância e tempo para chegar ao local do novo evento. Por exemplo, se um recurso está a caminho de um evento com o nível máximo de prioridade distante 10km da sua posição atual, a distância para o novo evento não deverá ser a distância da posição atual para a localização do novo evento, mas sim 10km mais a distância do evento atual para o novo evento. No entanto se o novo evento tiver uma prioridade inferior já deverá ser a distância entre a posição atual e a do novo evento a ter em conta para a classificação do recurso.

Este critério irá influenciar a distância e tempo para atingir o local de um novo evento, dependendo das prioridades dos eventos da atualmente atribuídos ao recurso.

### **Recurso vencedor**

Tendo em conta os critérios descritos, para identificar o recurso a atribuir o evento, será necessário correlacionar os vários parâmetros. Para a correlação

destes parâmetros foi adaptada a fórmula Euclidiana para o cálculo da distância entre dois pontos usando três dimensões:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3)$$

Onde "d" é a distância entre os dois pontos, e "x", "y" e "z" representam as três dimensões. Neste caso, as três dimensões usadas são: Tempo, Carga de Trabalho e Perfil.

Aplicando diretamente a formula e usando as mesmas dimensões para os vários critérios, a fórmula usada para a classificação do recurso fica:

$$Sc_r = \sqrt{\left(\frac{t_r}{t_{\max}}\right)^2 + \left(\frac{l_r}{l_{\max}}\right)^2 + \left(\frac{p_r}{p_{\max}}\right)^2} \quad (4)$$

Onde  $Sc_r$  é a classificação do recurso,  $t_r$  o tempo que o recurso leva a chegar ao local,  $t_{\max}$  o tempo máximo para chegar a um local no sistema,  $l_r$  a carga de trabalho do recurso,  $l_{\max}$  carga máxima,  $p_r$  o perfil do recurso e  $p_{\max}$  como valor máximo para o perfil. Para simplificação, é considerado 30 minutos como tempo máximo para alcançar o local de um evento e 5 eventos como carga de trabalho máxima do recurso e 5 como perfil máximo.

O recurso a que será atribuído o evento será o que tiver a classificação mais baixa:

$$W_r = \min\{Sc_1, Sc_2, \dots, Sc_n\} \quad (5)$$

### **Reatribuição de eventos**

Se o recurso tiver que interromper o tratamento de um evento, por atribuição de um novo evento com mais prioridade, será executada uma nova classificação para o evento que ficou suspenso. Este procedimento será repetido para todos os eventos suspensos até que todos fiquem atribuídos.

#### 4.4.2 Processo de Seguimento de Intrusos

Em caso de ser detetado um evento de Intrusão por uma câmara, o sistema fará o seguimento do suspeito dentro da sua área de atuação. Este processo de seguimento de intrusos, resumido na Figura 4.18, é iniciado quando uma câmara deteta um movimento e notifica o sistema com a imagem do primeiro movimento detetado, data/hora inicial e coordenadas da deteção. Ao mesmo tempo é iniciada a gravação do vídeo, que será enviado posteriormente.

É então criado no sistema um evento do tipo Intrusão, atribuída uma ação do tipo Seguimento e atribuído a um recurso disponível no sistema para o seu tratamento. Ao recurso que ficar atribuído para o tratamento do evento, será enviada a imagem e vídeo da intrusão para confirmação do evento. O recurso poderá também aceder ao vídeo em direto caso pretenda mais informação.

Caso o evento seja confirmado, o Agente Ação pesquisa a existência de outros eventos do tipo Intrusão ativos no sistema, para estabelecer uma eventual correlação entre eles.

Caso exista outro evento ativo, é enviada para a câmara que detetou o último evento a lista de câmaras mais próximas que detetaram eventos deste tipo, para poderem ser feitas comparações de “suspeito” entre elas. Após a comparação entre “suspeitos” é atribuído um grau de certeza entre 0 e 1 a cada comparação. Caso o grau de certeza seja maior do que 0,5 é criada uma relação lógica entre os eventos associados às intrusões e notificado o recurso para a possibilidade de existência de eventos correlacionados



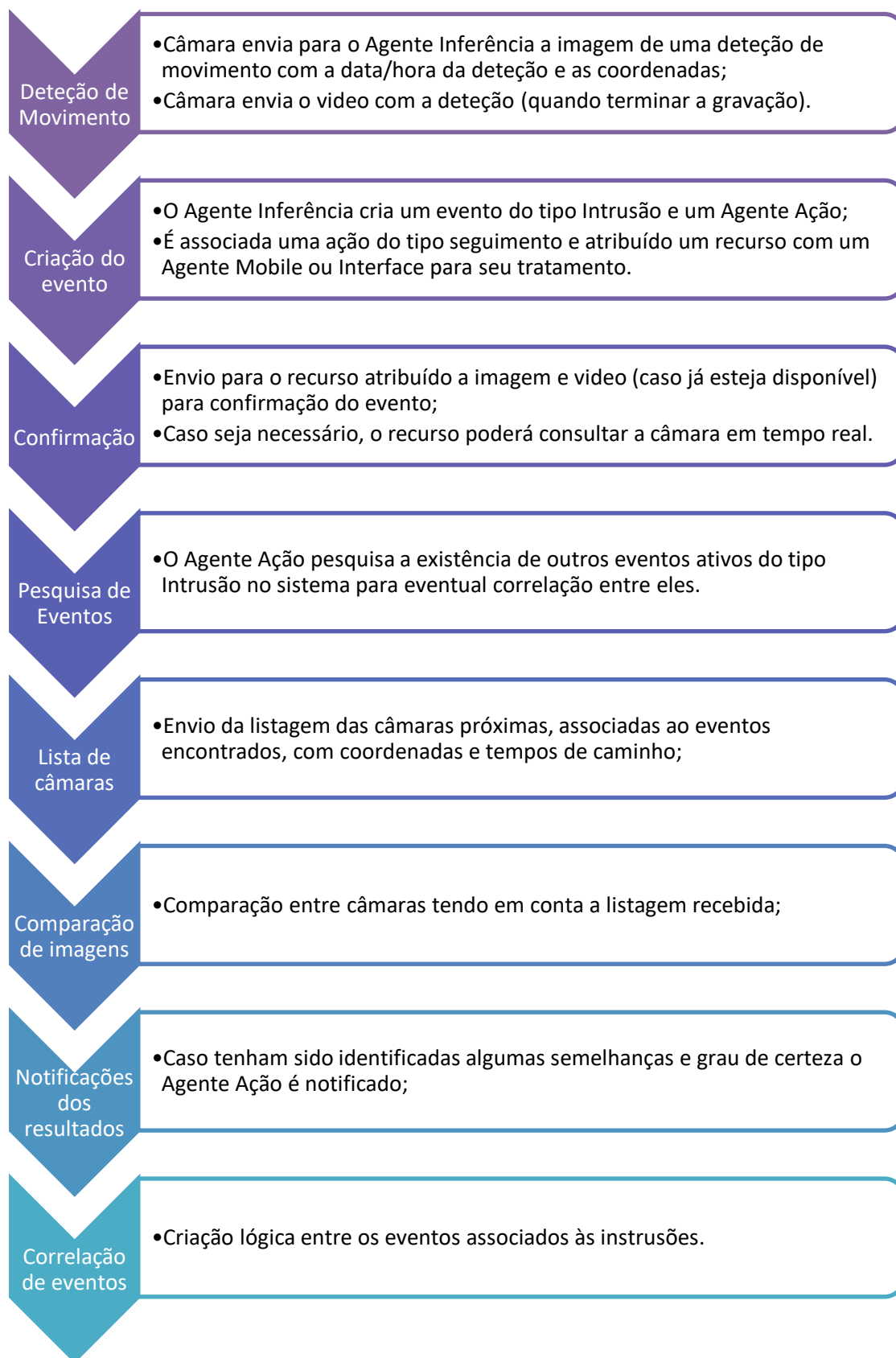


Figura 4.18 – Resumo do processo de Seguimento de suspeitos



## Implementação da arquitetura

Como descrito anteriormente, a arquitetura proposta é baseada em Agentes de Software, existindo várias abordagens para a sua implementação. Seguidamente é feita uma breve descrição de algumas dessas plataformas e da solução adotada para a implementação de um sistema baseado nesta arquitetura.

É também apresentado o protótipo desenvolvido que permitiu a realização de testes que possibilitaram validar os conceitos propostos.

### 5.1 *Plataforma de Agentes*

Considerando a opção de utilizar agentes de software foi necessário conduzir um estudo prévio de possíveis plataformas multiagentes, tendo sido identificadas as seguintes:

- **Zeus Agent Building Toolkit**<sup>9</sup> - O *Zeus Agent Building Toolkit* oferece ferramentas para a criação de aplicação com múltiplos agentes cooperativos. Implementa os mecanismos de comunicação FIPA e a integração com ontologias. A biblioteca foi desenvolvida pela Bri-

---

<sup>9</sup> <http://sourceforge.net/projects/zeusagent/>

*tish Telecommunications* mas encontra-se neste momento sem suporte, sendo ainda possível o seu *download*, mas a atividade do projeto e da comunidade nos últimos anos tem sido nula;

- **JACK Intelligent Agents**<sup>10</sup> - O JACK é um ambiente para a criação e execução de um sistema de agentes. É compatível com um conjunto variado de ambientes de execução (PC, Tablet, etc.). Oferece um conjunto de conceitos (*Capabilities, Events, Plans*, etc.) e de funcionalidades (Linguagens, mecanismos de comunicação, etc.) que permite o desenvolvimento de agentes colaborativos. É desenvolvido pela *Agent Oriented Software Limited* e possui um conjunto de informação de suporte considerável e desenvolvido. Possui um sistema de suporte pago, mas também uma comunidade de acesso gratuito<sup>11</sup> mas com uma atividade relativamente reduzida;
- **JADE**<sup>12</sup> - JADE é uma *framework* criada pela *Telecom Itália* para implementação de sistemas multiagentes e que implementa as normas definidas de FIPA. Implementada totalmente em JAVA, a sua arquitetura permite que seja distribuída por diferentes computadores em paralelo e em diferentes sistemas operativos. Oferece todos os mecanismos de gestão e comunicação entre agentes e um mecanismo de suporte para ontologias. A plataforma base é complementada com o *add-on LEAP* que permite a execução de agentes em ambientes móveis JAVA (exemplo: telemóvel ou tablet) com um impacto reduzido no desempenho dos dispositivos, totalmente compatível e

---

<sup>10</sup> <http://www.agent-software.com.au/products/jack/>

<sup>11</sup> <http://tech.groups.yahoo.com/group/JACKagents/>

<sup>12</sup> <http://jade.tilab.com/>

com um desenvolvimento transparente para o programador. É considerada atualmente a *Framework* de referência para sistemas de agentes e possui uma vasta comunidade que possui uma forte e constante atividade.

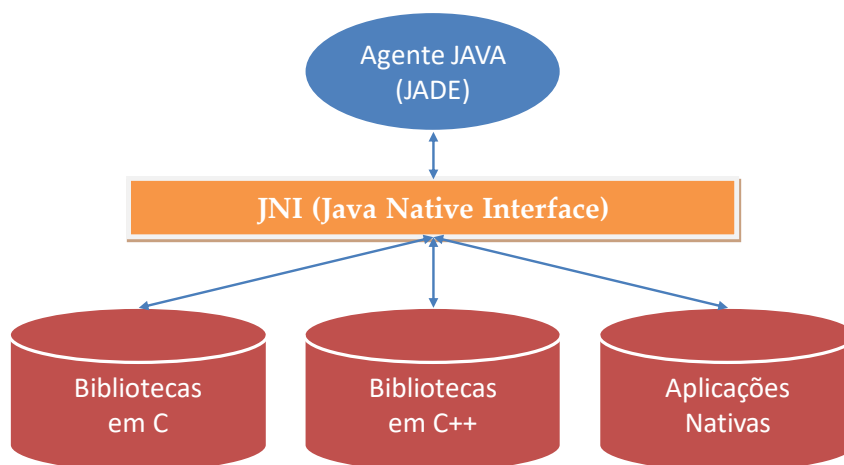
A análise às plataformas existentes permitiu concluir que as plataformas JADE e LEAP são a referência para o desenvolvimento de agentes, existindo um desenvolvimento contínuo de novas funcionalidades e mantendo uma comunidade de discussão e partilha muito alargada e com grande atividade. Estes fatores, aliados à experiência na utilização de JADE na implementação de projetos anteriores, levaram à decisão de utilização do JADE e LEAP para o desenvolvimento do sistema.

## 5.2 *Integração de sensores*

A pluralidade da informação recebida pelo sistema é a base de todo o seu funcionamento, no entanto, a sua integração é uma das questões mais complexas. A arquitetura descrita prevê a entrada de novos sensores no sistema, sendo que a criação dos Agentes Sensor deve ser automática, assim como a dos Agentes Processador.

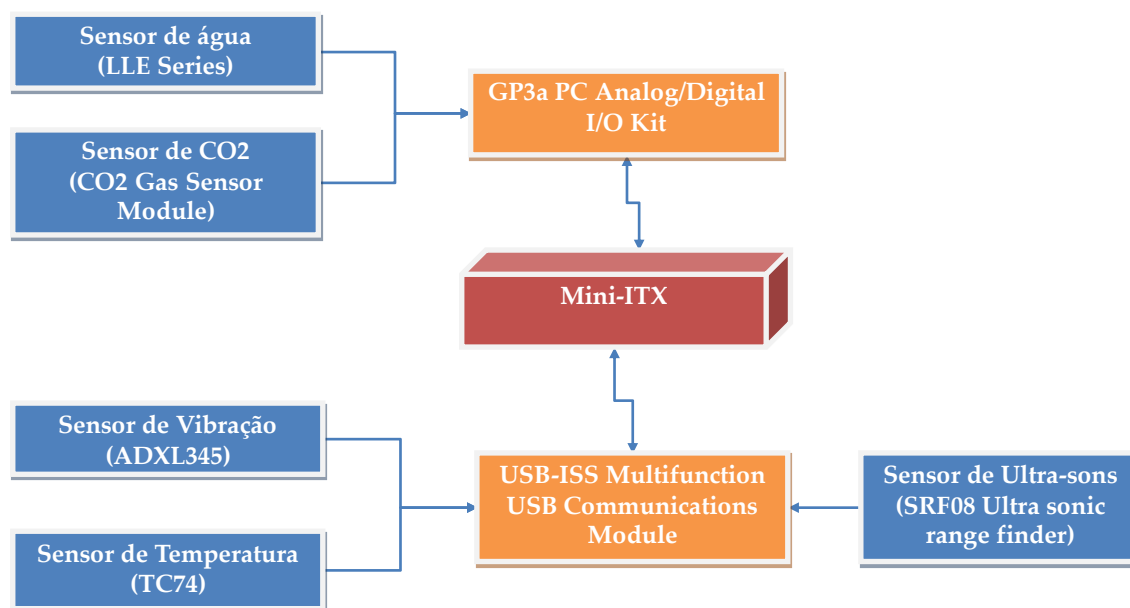
A integração de novos sensores no sistema, dado a diversidade de dispositivos existentes no mercado assim como as diferentes formas em que os sensores disponibilizam a informação, é um dos principais obstáculos para uma perspectiva *Plug-and-play* de integração de sensores.

De forma a agilizar a integração de sensores, permitindo uma abordagem *Plug-and-play*, foi desenhada uma camada de software que permite a abstração do tipo de sensor e da forma como este disponibiliza a informação. Esta camada permite a ligação entre as bibliotecas dos sensores e a plataforma de agentes, disponibilizando os métodos desenvolvidos em C através de classes Java, usadas pelos Agentes Sensor.



**Figura 5.1 – Comunicação Agentes - Sensores**

Para o desenvolvimento desta camada foi utilizada a *Java Native Interface* (JNI). A JNI é uma plataforma que permite executar código numa *Java Virtual Machine* (JVM) para chamar classes desenvolvidas em aplicações nativas ou bibliotecas desenvolvidas em outras linguagens como C, C++ ou Assembly.



**Figura 5.2 – Diagrama de sensores**

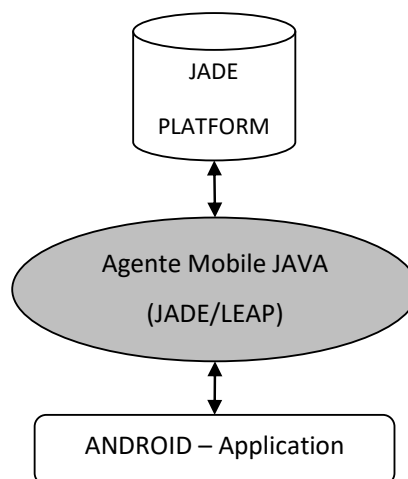
Com esta abordagem para aquisição de dados ao nível de software, é possível usar diferentes tipos de conexões com os sensores em termos de hardware, por exemplo uma placa de aquisição de dados ou através de USB. Na Figura 5.2 é ilustrado um exemplo de integração de vários tipos de sensores com diferentes tipos comunicações, sendo o sensor de água e o sensor de CO2 ligados a

uma placa de aquisição de dados e os sensores de Vibração, Temperatura e Ultra-sons através de UBS usando uma conversão USB-ISS que permite a comunicação em I2C.

### 5.3 *Sistema de Agentes*

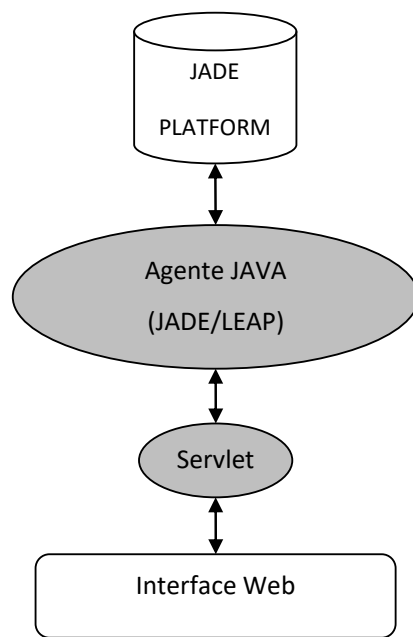
Após a escolha do JADE e LEAP para o ambiente de desenvolvimento foi efetuada a instalação das plataformas e prova de conceito da arquitetura proposta para o sistema. Para a realização da prova de conceito foram implementados agentes emuladores de sensores, de processadores e de inferência e definidos mecanismos de comunicação e de interação entre eles. Estes desenvolvimentos permitiram a demonstração da prova de conceito, servindo de base para o desenvolvimento do sistema.

Foram também efetuados testes de interação entre o JAVA e o LEAP, utilizando um telemóvel HTC, tendo sido atingidos os objetivos dos testes com sucesso. Esta interação permitiu desenvolver clientes para telemóveis com sistema Android, sendo assim possível a consulta de dados e interação com o sistema.



**Figura 5.3 – Módulo mobile**

Para o acesso web ao sistema foi também desenvolvido um módulo que permite a comunicação com o sistema de agentes JAVA, através de uma *servlet*.



**Figura 5.4 – Módulo Interface**

Foi realizado também um estudo de identificação de *frameworks* de inferência, tendo sido criada uma lista com os sistemas a analisar mais detalhadamente para utilização no agente de inferência. Foram identificadas as seguintes *frameworks*:

- **Drools**<sup>13</sup> - O Drools é um motor de inferência implementado com base no algoritmo Rete e adaptado para JAVA. A adaptação do Rete ao paradigma *object-oriented* permite a definição de regras de inferência mais abrangentes, tendo em conta os objetos de negócio. Apesar de estar programado em JAVA também pode ser executado em .NET. A sua implementação permite também que se ligue a linguagens definidas pelo utilizador. Os conjuntos de regras podem atualmente ser escritos em Java, Python ou Groovy. Um fator decisivo é a capacidade que oferece para *Declarative Programming* sendo

---

<sup>13</sup> <http://www.jboss.org/drools>



flexível na adaptação à semântica do problema, recorrendo a Domain Specific Languages (DSL) definidas através de XML;

- **R**<sup>14</sup> - O R é uma linguagem e plataforma para computação estatística. Apesar de estar mais direcionado para operações de estatística, oferece também um conjunto variado de classificadores como Árvores de decisão ou vizinhos-mais-próximos (*nearest neighbours*);
- **Weka**<sup>15</sup> - O Weka oferece uma abrangente coleção de algoritmos de *machine learning*, podendo ser executados diretamente através do software disponibilizado ou integrado diretamente em código JAVA. Entre muitas outras funcionalidades são disponibilizadas várias ferramentas para pré-processamento, classificação e regressão;
- **RapidMiner**<sup>16</sup> - À semelhança do Weka, o RapidMiner oferece uma alargada escolha de algoritmos de *machine learning*. Pode também ser executada através da aplicação disponibilizada ou integrada no código produzido. Oferece também entre muitas outras a possibilidade de integração de dados, ETL (Extract, Transform and Load) e de análise de dados.
- **JESS** – é um ambiente de desenvolvimento de scripts e motor de regras desenvolvido em Java. Permite desenvolver software Java com capacidade de decisão, usando um conjunto de regras definidas. JESS é pequeno, leve e um dos mecanismos mais rápidos de regras disponíveis. A sua linguagem permite acesso a todas as APIs

---

<sup>14</sup> <http://www.r-project.org/>

<sup>15</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

<sup>16</sup> <http://rapid-i.com/>

Java. O JESS usa uma versão melhorada do algoritmo Rete às regras do processo.

Após a análise e testes efetuados foi selecionado o Drools como motor de inferência para as regras definidas no sistema. A adoção deste motor permite a definição de vários tipos de regras usadas no sistema. O sistema permite a definição de regras para detecção de eventos.

As regras representam condições definidas pelos utilizadores segundo os valores dos parâmetros, por exemplo, “temperatura maior que 50°C”. As regras definidas são a base para o motor de inferência, que constantemente avalia os valores dos parâmetros e dispara eventos caso as condições se verifiquem. Neste sistema, adaptado para permitir novos tipos de regras/condições (como por exemplo as regras Temporais), podem ser definidas regras que contenham:

- Condições de comparação (maior, menor, igual...) entre parâmetros ou entre parâmetros e valores;
- Condições lógicas (AND, OR, NOT...);
- Condições Temporais (durante “X” tempo, enquanto...);
- Regras (usar outra regra como parâmetros).

Um exemplo de uma regra simples (formato legível):

IF (Parâmetro1 > 30) THEN EVENTO\_INCENDIO

Um exemplo de uma regra temporal (formato legível):

IF (Parâmetro1 > 20 AND Parâmetro2 = 5) (DURING 120 SECONDS)  
THEN EVENTO\_ INCENDIO

Para edição das regras foi desenvolvida uma interface gráfica que permite a construção de regras usando os parâmetros existentes no sistema.

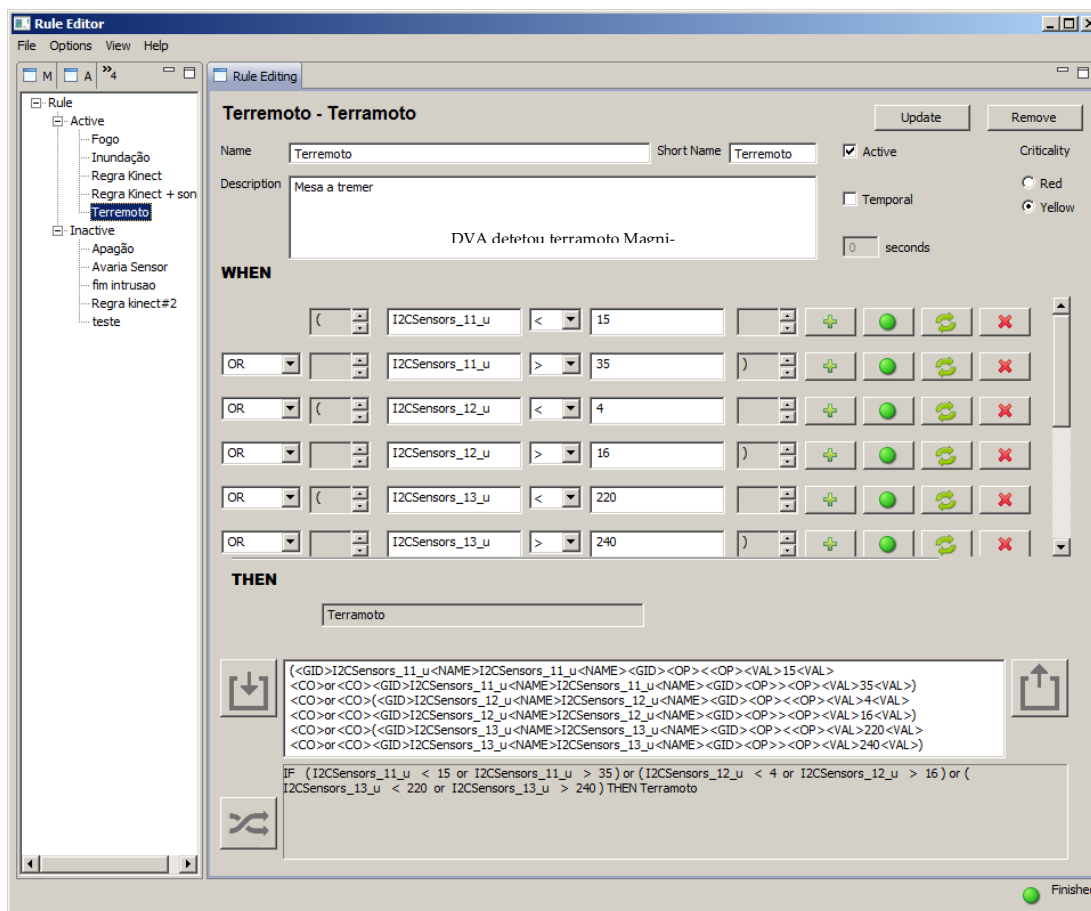


Figura 5.5 – Editor de regras

## 5.4 Protótipo desenvolvido

O protótipo desenvolvido permitiu observar os conceitos desenvolvidos e validar as melhorias esperadas com a implementação da nova arquitetura.

Este protótipo foi composto por diferentes módulos que representaram as várias componentes da arquitetura:

- Interface web (Agente Interface);
- Aplicação mobile (Agente Mobile);
- Sensores (Agentes Sensor e Processador);
- Processamento (Agentes Inferência, Ação, Backup e Monitorização).

### 5.4.1 Interface web

A interface web desenvolvida permite a interação dos utilizadores com todo o sistema, assim como a configuração das suas várias componentes. Para além da interface web disponibilizada para qualquer browser, este módulo inclui um Agente Interface que comunica com o restante sistema.

Através desta interface é possível consultar no mapa, por tipo e área, os Eventos, Sensores e Parâmetros. Para cada Evento está disponível: a sua localização, estado, descrição, anexos, recursos atribuídos, parâmetros e sensores associados, eventos relacionados e histórico de ações (em forma de lista e ou linha temporal). Para os Sensores é possível consultar a sua localização, detalhe e valor atual dos parâmetros associados. No caso dos Parâmetros, para além da sua localização é possível consultar valor atual e detalhe.

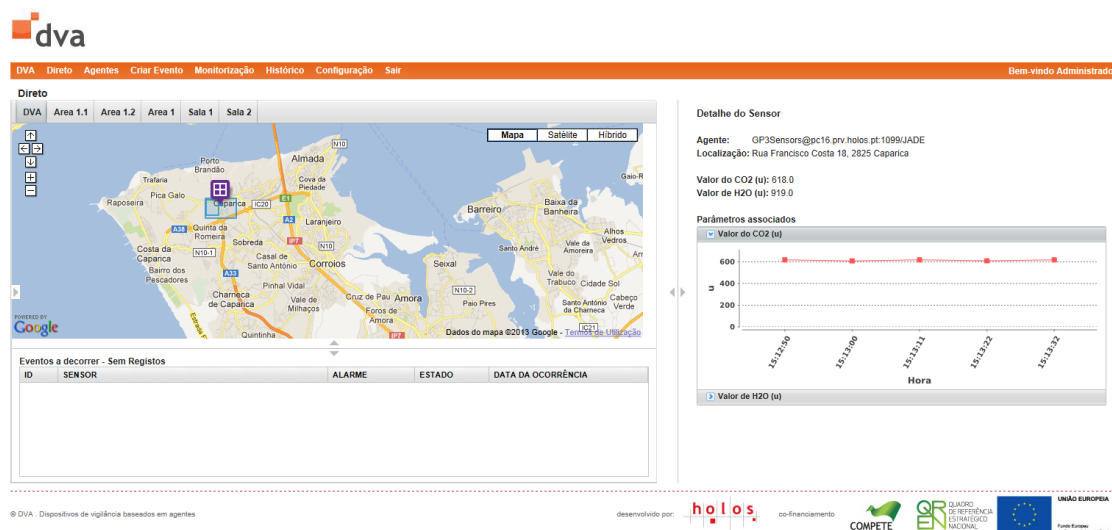


Figura 5.6 – Exemplo de visualização do detalhe de um sensor

Na interface web é possível criar eventos, especificando a sua localização, tipo e criticidade. Para além destas caracterizações do evento, é possível incluir uma pequena descrição do mesmo e adicionar anexos, tais como imagens e vídeos.

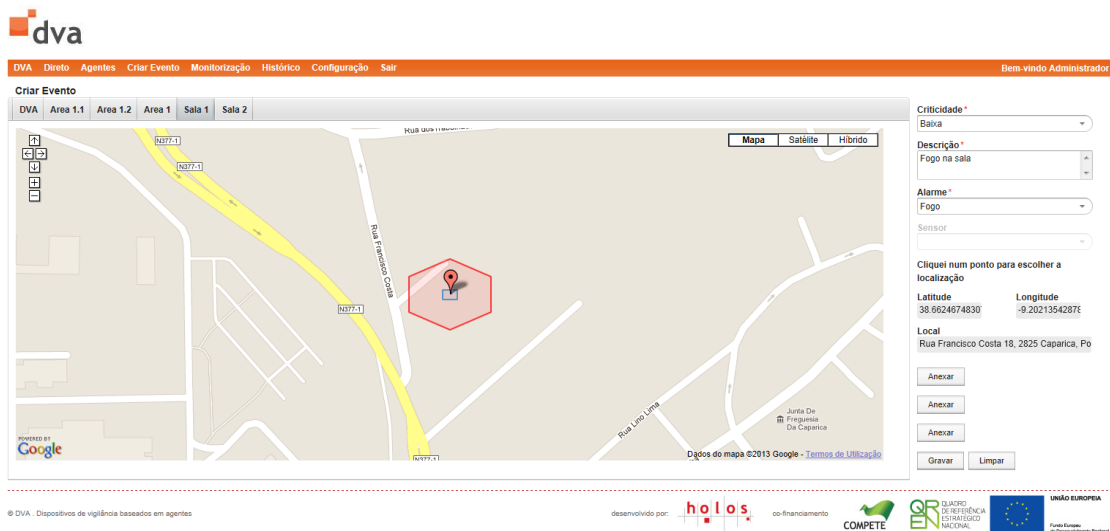


Figura 5.7 – Exemplo da criação de um evento

Outra das funcionalidades disponíveis, é a gestão/tratamento de eventos, permitindo aos utilizadores:

- Confirmar/Concluir/Cancelar Eventos;
- Atribuição/desafetação do evento a determinado recurso;
- Adicionar anexos (imagens e vídeos);
- Relacionar com outro evento;
- Criar evento relacionado;

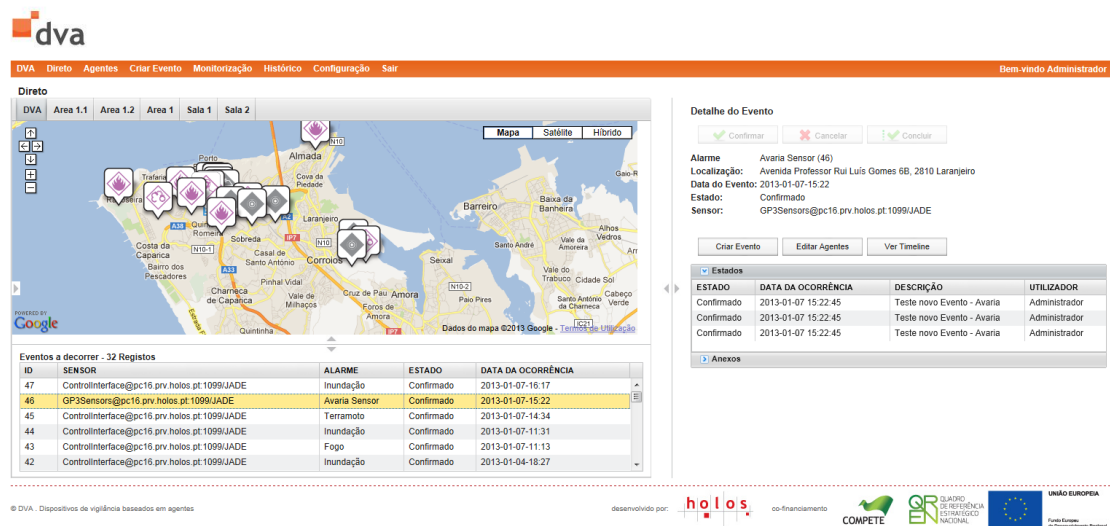
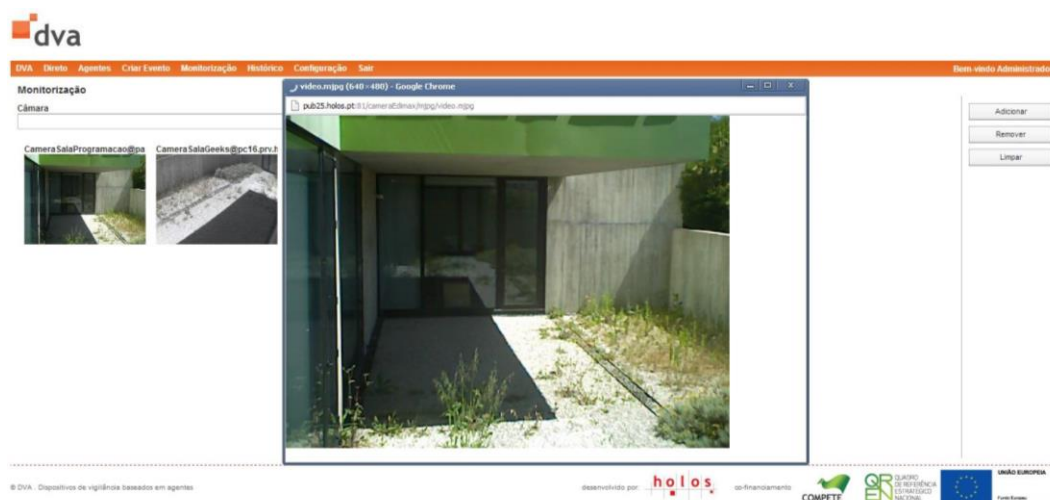


Figura 5.8 – Exemplo detalhe de um evento

Caso estejam incluídas câmaras no sistema, é possível a visualização em tempo real e em simultâneo do vídeo destas câmaras.



**Figura 5.9 – Exemplo monitorização de câmaras**

Para além da visualização de toda a informação em tempo real, como descrito, a interface web também disponibiliza uma consulta do histórico, onde se pode consultar o histórico de:

- Eventos, por tipo, por data, por estado e área: localização, estado, detalhe, anexos, recursos atribuídos, tratamento, parâmetros e sensores associados, eventos relacionados e histórico de ações (por lista e linha temporal);
- Sensores, por estado, por tipo, por evento e área: localização, detalhe e valor dos parâmetros associados (por gráfico);
- Parâmetros, por tipo, por evento e área: localização, detalhe e valor (por gráfico);
- Logs, por data e por utilizador;
- Visualização das Gravações agendadas, por data e câmara;
- Linha temporal de eventos, por tipo, por estado e data;



**Figura 5.10 –Exemplo de visualização de parâmetros e eventos**

Por último, neste módulo é disponibilizada a configuração de algumas componentes do sistema, tais como:

- Definição de categorias dos utilizadores;
- Criação de utilizadores, com configuração de categorias, perfis, áreas e acessos;
- Criação de parâmetros calculados: definição de novos parâmetros, utilizando parâmetros já existentes, podendo ser definidas fórmulas para o seu cálculo;
- Raio de ação dos parâmetros;
- Criação de ações para os diferentes tipos de eventos (emails, sms, atribuição do evento por estado, área, por tipo de utilizador e por perfil;
- Definição de áreas;
- Calendarização de gravação de vídeos, por data e câmara;
- Definição dos feriados.

### 5.4.2 Aplicação mobile

Para o acesso ao sistema através de telemóveis por parte dos utilizadores, permitindo assim a sua mobilidade (não tendo de estar em frente a um computador para consultar o sistema) e georreferenciação (usada pelo sistema na atribuição de eventos), foi desenvolvida uma aplicação mobile que permite o acesso ao sistema em plataformas *Android*.



Figura 5.11 – Módulo Mobile – Ecrã inicial



Figura 5.12 – Módulo Mobile – Consulta do mapa

Para além da interface com o utilizador, esta aplicação também implementa o Agente Mobile que comunica com o sistema.

Esta aplicação desenvolvida a pensar no acesso rápido ao sistema (ilustrado na Figura 5.11), quer para reportar um evento, quer para tratar um evento que seja atribuído ao utilizador, permite também consultar no mapa os Eventos, Sensores e Parâmetros ativos nesse momento (Figura 5.12).

Na criação de um novo evento (Figura 5.13), é possível fazer a caracterização do evento, escolhendo o seu tipo e localização (automática ou escolhida



através do mapa). Tal como na interface web, é possível também adicionar uma descrição e anexos ao evento.

Em relação à notificação de um novo evento, o utilizador recebe uma notificação no telemóvel, permitindo assim ser alertado e aceder à aplicação para consulta do evento. A aplicação disponibiliza também a lista de eventos ativos, recebidos e enviados pelo utilizador (Figura 5.14). Esta lista de eventos pode ser ordenada por diversos tipos de critérios, como por exemplo, prioridade, localização ou tipo.



Figura 5.13 – Módulo Mobile – Criação de um evento

Activos	Recebidos	Enviados
	infAgent Normal 2012-01-12 18:23:49	
	infAgent Normal 2012-01-12 18:23:46	
	infAgent Normal 2012-01-12 18:23:27	
	infAgent Normal 2012-01-12 18:23:22	
	infAgent Crítico 2012-01-12 18:22:29	
	infAgent Normal 2012-01-12 18:03:45	
	infAgent Crítico 2012-01-12 18:03:45	

Figura 5.14 – Módulo Mobile – eventos atribuídos

### 5.4.3 Sensores

Na construção do protótipo foram implementados diversos tipos de sensores, tais como: temperatura; ultrassons; água; luminosidade; vibração (acelerómetro); humidade; Kinect (usado para a deteção de pessoas); Incêndio; CO2. Esta diversidade de sensores permitiu testar vários tipos de comunicações entre o sistema e os sensores, tendo sido construídas pequenas plataformas de hardware para proceder à recolha de dados destes sensores, tal como descrito na secção “5.2 - Integração de sensores”.

Nas imagens da Figura 5.15 está ilustrado um exemplo destas pequenas plataformas de hardware que permitem a recolha de dados. Neste caso, a caixa recolhe valores de temperatura, luminosidade e distância (ultrassons) e como saída para o sistema apresenta um único cabo USB que pode ser ligado a qualquer computador onde será executado Agente Sensor.



**Figura 5.15 – Exemplo 1 de montagem DVAs**

Para a aquisição de dados e execução dos Agentes Sensor e Processamento, foram usados diferentes tipos de soluções, desde soluções muito baratas, como a Raspberry Pi<sup>17</sup>, outras com um pouco mais de capacidade de processamento, como Mini-ITX, e os tradicionais Computadores Desktop. Na Figura 5.16 é ilustrado um exemplo de uma caixa que, para além do hardware de recolha dos sensores, continha também uma Raspberry Pi em que eram executados os respetivos Agentes Sensor e Processamento.

---

<sup>17</sup> <https://www.raspberrypi.org/>



Figura 5.16 - Exemplo 2 de montagem de DVAs

#### 5.4.4 Processamento

A implementação desta arquitetura baseada em multiagentes, permite a adoção de computadores de baixo custo com pouca capacidade de processamento, visto que a execução de cada agente está repartida por diversos computadores. Assim, o sistema pôde ser implementado em computadores de baixo custo e com pouca capacidade de processamento como Raspberry Pi, Mini-ITX e Computadores Desktop, sem a necessidade do uso de servidores com grande capacidade de processamento.

Em resumo, foram usados no protótipo:

- 2 Raspberry Pi – recolha de dados sensoriais e execução dos respectivos Agentes Sensor e Processamento;
- 1 Mini-ITX – execução dos Agentes Backup, Monitorização e outros Agentes Sensor e Processamento (que não estejam ligados a Raspberry Pi);

- 2 Computadores Desktop - execução dos Agentes Interface (juntamente com a interface web), Inferência e Ação;
- 5 Telemóveis Android – execução dos Agentes Mobile (com a aplicação mobile).

Os computadores onde são executados os Agentes Backup, Interface, Inferência, Ação e Monitorização, podem ser alterados no decorrer do tempo, uma vez que os agentes podem mudar de computador. Caso o computador onde estavam a ser executados deixe de funcionar, o sistema deteta e cria o agente em outro computador que esteja disponível.

# 6

## Testes e resultados

Para validar o sistema desenvolvido, baseado na arquitetura proposta, foi necessário realizar vários testes e analisar os seus resultados. Assim, foram executados testes em ambiente simulado e em ambiente real. Neste capítulo são descritos alguns destes testes.

### 6.1 *Cenário*

O cenário definido para os testes foi baseado numa área onde seria necessário a monitorização de pontos críticos. Os locais a monitorizar, utilizados nos testes, estão identificados por L1, L2, ...Ln. Na Figura 6.1, é ilustrado este cenário através de um mapa e onde estão exemplificados 5 locais.

Neste cenário, e como o objetivo era testar sobretudo a capacidade do sistema em fazer uma gestão otimizada dos recursos (humanos), tendo em conta a sua posição geográfica, cada local só tinha uma área a monitorizar (exemplo uma sala, ou um espaço exterior).



**Figura 6.1 – Identificação dos locais**

Paralelamente, para cobrir as várias vertentes que a arquitetura abrange, foram instalados vários tipos de sensores (identificados por S1, S2, ... Sn.) em cada um dos locais, assegurando assim, a simulação de deteção de intrusão, terremotos, gás, incêndios e inundação. A configuração dos vários locais, em termos de sensores está resumida na Tabela 6.1.

Local	Equipamentos
<b>L1</b>	S1 - Caixa I2C (Sonar e Luminosidade) S2 - Caixa GP3 (Incêndio e Água) S3 - Camera Edimax (Movimento, Humanos, Incêndio)
<b>L2</b>	S4 - Caixa I2C+GP3 (Temperatura, Sonar, Luminosidade, Incêndio, Dióxido de Carbono (CO2), Água (H2O))
<b>L3</b>	S5 - Camera Acti (Movimento, Humanos, Incêndio) S6 - Caixa I2C (Temperatura, Sonar e Luminosidade)
<b>L4</b>	S7 - Caixa I2C (Temperatura e Acelerómetro)
<b>L5</b>	S8 - Caixa I2C e Raspberry (Humidade, Temp. e Acelerómetro)

**Tabela 6.1 – Distribuição dos sensores por local**

Tendo em conta os locais definidos, foram configuradas no sistema o mesmo número de áreas (uma por local) usando o alcance dos vários sensores presentes em cada local para definir a delimitação das áreas.

Em cada local foram instalados vários sensores, o que permitiu a geração de diversos tipos de parâmetro. Os parâmetros foram identificados com a grandeza que representam (ex.: temperatura) concatenada com a área a que pertencem (ex.: temperaturaL1), facilitando assim a leitura das regras posteriormente definidas. As grandezas disponíveis, através dos sensores instalados em cada uma das áreas, são:

- **Distância** representa a distância do sensor(sonar) à parede contrária em cm - permitindo detetar intrusões, caso esta distância se altere por alguém estar entre o sonar e a parede;
- **Luminosidade** quantifica a luminosidade em lumens na área - permitindo detetar alterações bruscas de luz, como por exemplo, uma lanterna durante a noite;
- **Incêndio** representa a presença ou não de chamas, podendo ser detetado por um sensor específico ou pelas câmaras – usado para a deteção de incêndios;
- **Movimento** representa a deteção pelas câmaras de movimento (0 – sem movimento e 1 movimento) – usado na deteção de intrusos;
- **Humanos** representa a deteção pelas câmaras de humano (0 – humano não detetado e 1 detetado pelo menos um humano) – usado na deteção de intrusos;
- **Temperatura** apresenta a temperatura na área em graus centígrados – pode ser usado também na deteção de incêndios detetando subidas de temperatura definidas como fora do normal;

- **Água** revela presença de água – usado na detecção de inundações;
- **CO2** representa a detecção de dióxido de carbono na área (0 não detectado e 1 detectado) – usado para a detecção de incêndios identificando a presença de níveis elevados de CO2;
- **Acelerómetro** quantifica as alterações de posição do sensor (x, y e z) – podendo ser usado para a detecção de terremotos ou para intrusões quando acoplado a uma entrada (ex.: uma porta ou janela);
- **Humidade** específica a percentagem de humidade na área – este parâmetro pode ser usado na detecção de inundações, detetando variações bruscas de humidade na área.

Na Tabela 6.2 são listados alguns dos parâmetros disponíveis para consulta no sistema e para a elaboração de regras.

Área	Parâmetros		
<b>L1</b>	DistanciaL1	IncendioL1	C_MovimentoL1
	LuminosidadeL1	AguaL1	C_IncendioL1
			C_HumanoL1
<b>L2</b>	TemperaturaL2	LuminosidadeL2	AguaL2
	DistanciaL2	CO2L2	IncendioL2
<b>L3</b>	DistanciaL3	C_HumanoL3	C_IncendioL3
	LuminosidadeL3	C_MovimentoL3	
<b>L4</b>	TemperaturaL4	AcelerometroL4	
<b>L5</b>	TemperaturaL5	HumidadeL5	AcelerometroL5

**Tabela 6.2 – Exemplo de parâmetros definidos**

Com base nos parâmetros existentes, foram definidas várias regras. Na Tabela 6.3, estão exemplificadas três destas regras, em que cada uma das regras



foi especificada para uma área diferente, tendo em conta a localização dos sensores.

Nome da Regra	Área	Parâmetros Usados	Regra	Tipo de evento
<b>Incêndio L1</b>	L1	IncendioL1 C_IncendioL1	IF (IncendioL1 = 1  OR  C_IncendioL1 = 1)  THEN EVENTO_INCENDIO	Incêndio
<b>Inundação L2</b>	L2	AguaL2	IF (AguaL2 = 1)  THEN EVENTO_INUNDACAO	Inundação
<b>Intrusão L3</b>	L3	DistanciaL3 C_HumanoL3	IF (DistanciaL3= 1  OR  C_HumanoL3 = 1)  THEN EVENTO_INTRUSAO	Intrusão

**Tabela 6.3 – Exemplo de regras definidas**

Paralelamente, associadas a cada tipo de evento, foram definidas ações para cada um dos estados em que estes eventos podem estar. As ações disponíveis eram: Envio de emails, envio SMS e atribuição do evento a um recurso para confirmação e para tratamento do mesmo.

De forma a executar os testes com todos os tipos de eventos e todos os perfis dos Agentes Mobile/Interface, foram definidas ações para cada um dos tipos de evento e não foram limitados os perfis para a atribuição do evento. As ações definidas foram semelhantes a:

- Eventos tipo Incêndio:
  1. Enviar email de notificação em cada um dos estados;
  2. Enviar sms no estado “Confirmado”;

3. No estado “Por Confirmar” - atribuir o evento a um Agente Mobile; se falhar (por não conseguir encontrar nenhum Agente dentro do tempo definido), atribuir a um Agente Interface;
4. No estado “Confirmado” - atribuir o evento a um Agente Mobile; se falhar (por não conseguir encontrar nenhum Agente dentro do tempo definido), atribuir a um a Agente Interface;

## 6.2 Simulações

Através do cenário definido, foi possível realizar várias simulações, sobretudo tendo em vista a resposta do sistema em termos de Carga de Trabalho, Perfil e Distância ao evento dos recursos disponíveis.

Estas simulações foram realizadas sobre o sistema desenvolvido, sendo que as localizações foram pré-definidas em vez de obtidas dinamicamente por GPS, e os eventos foram inseridos por *script* no sistema.

Nesta seção são descritas algumas destas simulações e resultados obtidos, sendo que, o conjunto selecionado permite ilustrar o comportamento do sistema. Para além das simulações apresentadas, foram efetuadas diversas simulações em escala superior com resultados equivalentes, porque os algoritmos são de complexidade linear.

Para além do descrito no cenário, foram definidos, para as várias simulações, um conjunto de recursos ligados ao sistema através de Agentes Mobile (descritos na Tabela 6.4) com diferentes perfis e localizações.

Este conjunto de recursos permitiu simular a resposta do sistema para o caso em que existe sempre pelo menos um recurso disponível para tratar um novo evento e para o caso em que existem mais eventos do que recursos disponíveis.

Para as várias simulações, foram definidos diversos tipos de eventos, podendo estes serem de diversos tipos e com diferentes prioridades.

Recurso	Perfil	Coordenadas (Lat,Lon)
A1	Polícia	"38.68316,-8.9103"
A2	Bombeiro	"38.66601,-8.9655"
A3	Segurança	"38.63195,-8.9415"
A4	Proteção Civil	"38.64161,-8.90579"
A5	Polícia	"38.67807,-8.9415"
A6	Bombeiro	"38.66695,-8.9328"
A7	Proteção Civil	"38.66403,-8.9384"
A8	Segurança	"38.64761, -8.9316"
A9	Segurança	"38.66409, -8.8949"
A10	Polícia	"38.64804, -8.9730"

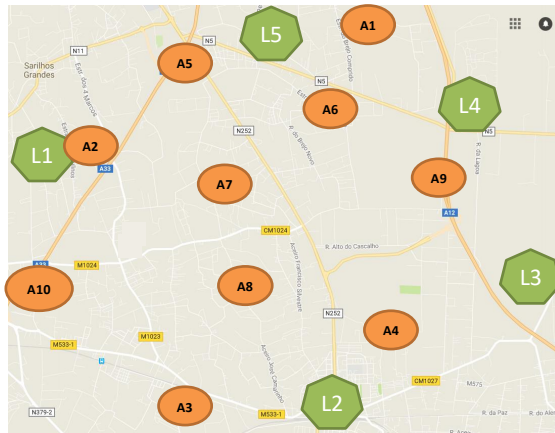
**Tabela 6.4 – Recursos usados nas simulações**

A localização dos eventos foi sempre baseada nos vários locais definidos no cenário. Na Tabela 6.5 estão descritos os vários eventos usados nas simulações.

Evento	Tipo	Prioridade	Coordenadas (Lat,Lon)
E1	Incêndio	Alta	"38.6459922,-8.885502"
E2	Inundação	Alta	"38.6645665,-8.9687895"
E3	Intrusão	Baixa	"38.6323012,-8.9158815"
E4	Gás	Média	"38.6761186,-8.8961279"
E5	Terremoto	Média	"38.6858942,-8.9210996"
E6	Inundação	Baixa	"38.6323012,-8.9158815"
E7	Gás	Média	"38.6459922,-8.885502"
E8	Terremoto	Alta	"38.6858942,-8.9210996"
E9	Incêndio	Alta	"38.6761186,-8.8961279"
E10	Intrusão	Baixa	"38.6645665,-8.9687895"

**Tabela 6.5 – Eventos usados nas simulações**

Na Figura 6.2, é ilustrada no mapa a localização inicial dos recursos definidos e os locais definidos.



**Figura 6.2 – Mapa com a localização dos eventos e recursos**

Como referido anteriormente, o objetivo principal das simulações aqui descritas, era a análise do sistema em termos de Carga de Trabalho, Perfil e Distância ao evento dos recursos. Desta forma, foram definidos alguns critérios e condições iniciais, tais como:

- Todos os recursos começaram as simulações com zero eventos atribuídos e encontravam-se sempre na posição inicial, definida na Tabela 6.4;
- Todos os recursos estavam com o meio de locomoção igual, de carro;
- Todas as distâncias/tempo ao evento foram obtidas através do mesmo serviço, Google Maps;
- Todas as ações definidas só pretendiam um recurso por evento;
- Todos os eventos estavam no estado confirmado;
- Todos os eventos considerados em cada uma das simulações foram recebidos ainda com o recurso na sua posição inicial;
- A ordem de chegada dos eventos nas simulações, assim como o momento da atribuição do evento, é designada por  $t_1, t_2, \dots, t_n$ , onde 1 é o primeiro acontecimento e  $n$  o último.

- Foi considerado que o tempo de tratamento dos eventos no local era nulo, sendo assim só considerado tempo que o recurso levou a chegar ao local.

### **6.2.1 Caso 1 – 10 Recursos, 10 Eventos**

#### **1ª Simulação**

Nesta simulação o objetivo foi testar a resposta do sistema tendo em conta só a distância do recurso ao evento e o perfil do recurso. Neste teste, o sistema não leva em consideração a lista de eventos de cada recurso no momento de atribuição do evento. Outra das limitações desta simulação é a posição usada para o cálculo da distância, que independentemente dos eventos atribuídos ao recurso, é sempre usada a posição inicial para a classificação do recurso. No entanto, na simulação foi contabilizado o tempo que os recursos levariam a tratar todos os eventos da sua lista. Resumindo, as condições particulares desta simulação foram:

- Não foi considerada a prioridade dos eventos;
- Distância ao evento sempre calculada relativamente à posição inicial;
- Não foi usada a Carga de Trabalho na classificação dos recursos.

#### **2ª Simulação**

Na 2ª simulação, para além da distância ao evento e perfil, foi considerada também a carga de trabalho do recurso. Embora tenha sido levada em conta a carga de trabalho na classificação dos recursos, não foi ainda considerado, na classificação do evento, o tempo que o recurso leva a tratar os eventos atribuídos, isto é, foi sempre considerada a posição inicial do recurso para o cálculo da distância.

Tal como na simulação anterior, também foi contabilizado o tempo que os recursos levariam a tratar todos os eventos da sua lista.

Resumindo, as condições particulares desta simulação foram:

- Não foi considerada a prioridade dos eventos;
- Distância ao evento sempre calculada relativamente à posição inicial;
- Foi usada a Carga de Trabalho na classificação dos recursos, no entanto não será levado em conta o percurso do recurso (tendo em conta a sua lista de eventos por tratar) para chegar ao evento.

### **3ª Simulação**

A 3ª simulação, para além da carga, perfil e distância ao evento, já considerou o tempo de tratamento dos vários eventos atribuídos ao recurso no momento da classificação. Tal como na simulação anterior, também foi contabilizado o tempo que os recursos levariam a tratar todos os eventos da sua lista. Resumindo, as condições particulares desta simulação foram:

- Não foi considerada a prioridade dos eventos;
- Distância ao evento calculada tendo em conta a lista de eventos do recurso;

### **4ª Simulação**

A 4ª simulação, simulou o sistema por completo, isto é, na classificação dos recursos usou a carga de trabalho, perfil e distância ao evento tendo em conta a lista de eventos do recurso e a prioridade dos eventos. Neste caso, se um dos eventos da lista do recurso tiver prioridade inferior à do novo evento, este não será considerado na classificação do recurso. Nesta simulação, não foi considerada a reordenação e reatribuição de eventos no caso de ser atribuído um evento mais prioritário que outros que existam na lista do recurso.

	t1			t2		t3		t4		t5		t6		t7		t8		t9		t10		t11		t12		t13		t14		t15		t16		t17		t18		t19		t20	
AG	CT	E1	L.E.	T.C.	CT	E2	L.E.	T.C.	CT	E3	L.E.	T.C.	CT	E4	L.E.	T.C.	CT	E5	L.E.	T.C.	CT	E6	L.E.	T.C.	CT	E7	L.E.	T.C.	CT	E8	L.E.	T.C.	CT	E9	L.E.	T.C.	CT	E10	L.E.	T.C.	
1ª Simulação	A1	0	0.77			0	0.80	0.00	0	0.55		0.00	0	0.66		0.00	0	0.66		0.00	0	0.79		0.00	0	0.77		0.00	0	0.66		0.00	0	0.66		0.00	0	0.56		0.00	
	A2	0	0.50			0	0.22	E2 2.80	0	1.11	E2 2.80		0	0.77	E2 2.80		0	0.59	E2 2.80		0	0.53	E2 2.80		0	0.75	E2 2.80		0	0.59	E2 2.80	0	0.52	E2 2.80	0	1.00	E2 2.80	0	1.00	E2 2.80	
	A3	0	0.73			0	0.89	0.00	0	0.47		0.00	0	1.04		0.00	0	1.02		0.00	0	0.84		0.00	0	0.91		0.00	0	1.02		0.00	0	0.90		0.00	0	0.56		0.00	
	A4	0	0.70			0	0.69	0.00	0	1.04		0.00	0	0.69		0.00	0	0.69		0.00	0	0.49		0.00	0	0.54		0.00	0	0.69		0.00	0	0.82		0.00	0	1.14		0.00	
	A5	0	0.66			0	0.68	0.00	0	0.36	E3 9.02		0	0.66	E3 9.02		0	0.64	E3 9.02		0	0.67	E3 9.02		0	0.66	E3 9.02		0	0.64	E3 9.02	0	0.66	E3 9.02	0	0.38	E3 9.02	0	0.38	E3 9.02	
	A6	0	0.31	E1 7.17		0	0.44	E1 7.17	0	1.04	E1 7.17		0	0.68	E1 7.17		0	0.48	E1,E5 34.05		0	0.33	E1,E5,E6 47.70		0	0.65	E1,E5,E6 47.70		0	0.48	E1,E5,E6 8 61.18	0	0.38	E1,E5,E6 8,E9 69.08	0	1.08	E1,E5,E6 8,E9 69.08	0	1.08	E1,E5,E6 8,E9 69.08	
	A7	0	0.66			0	0.50	0.00	0	1.04		0.00	0	0.54	E4 10.92		0	0.51	E4 10.92		0	0.50	E4 10.92		0	0.49	E4,E7 24.67		0	0.51	E4,E7 24.67	0	0.70	E4,E7 24.67	0	1.04	E4,E7 24.67	0	1.04	E4,E7 24.67	
	A8	0	0.65			0	0.85	0.00	0	0.46		0.00	0	0.92		0.00	0	0.90		0.00	0	0.83		0.00	0	0.84		0.00	0	0.90		0.00	0	0.76		0.00	0	0.49		0.00	
	A9	0	0.60			0	0.88	0.00	0	0.46		0.00	0	0.91		0.00	0	0.89		0.00	0	0.83		0.00	0	0.80		0.00	0	0.89		0.00	0	0.74		0.00	0	0.55		0.00	
	A10	0	0.72			0	0.64	0.00	0	0.39		0.00	0	0.76		0.00	0	0.73		0.00	0	0.69		0.00	0	0.72		0.00	0	0.73		0.00	0	0.76		0.00	0	0.30	E10 6.53	0	0.30

	t1			t2		t3		t4		t5		t6		t7		t8		t9		t10		t11		t12		t13		t14		t15		t16		t17		t18		t19		t20	
AG	CT	E1	L.E.	T.C.	CT	E2	L.E.	T.C.	CT	E3	L.E.	T.C.	CT	E4	L.E.	T.C.	CT	E5	L.E.	T.C.	CT	E6	L.E.	T.C.	CT	E7	L.E.	T.C.	CT	E8	L.E.	T.C.	CT	E9	L.E.	T.C.	CT	E10	L.E.	T.C.	
2ª Simulação	A1	0	0.77			0	0.80	0.00	0	0.55		0.00	0	0.66		0.00	0	0.66		0.00	0	0.79		0.00	0	0.77		0.00	0	0.66		0.00	0	0.66		0.00	0	0.56		0.00	
	A2	0	0.50			0	0.22	E2 2.80	1	1.11	E2 2.80	1	0.80	E2 2.80	1	0.62	E2 2.80	1	0.56	E2 2.80	1	0.56	E2 2.80	1	0.78	E2 2.80	1	0.62	E2,E8 15.95	2	0.66	E2,E8 15.95	2	1.08	E2,E8 15.95	2	1.08	E2,E8 15.95			
	A3	0	0.73			0	0.89	0.00	0	0.47		0.00	0	1.04		0.00	0	1.02		0.00	0	0.84		0.00	0	0.91		0.00	0	1.02		0.00	0	0.90		0.00	0	0.56		0.00	
	A4	0	0.70			0	0.69	0.00	0	1.04		0.00	0	0.69		0.00	0	0.69		0.00	0	0.49	E6 8.38	1	0.57	E6 8.38	1	0.72	E6 8.38	1	0.84	E6 8.38	1	1.16	E6 8.38	1	1.16	E6 8.38			
	A5	0	0.66			0	0.68	0.00	0	0.36	E3 9.02	1	0.69	E3 9.02	1	0.67	E3 9.02	1	0.70	E3 9.02	1	0.69	E3 9.02	1	0.67	E3 9.02	1	0.67	E3 9.02	1	0.67	E3 9.02	1	0.69	E3 9.02	1	0.43	E3 9.02			
	A6	0	0.31	E1 7.17		1	0.49	E1 7.17	1	1.05	E1 7.17	1	0.71	E1 7.17	1	0.52	E1,E5 34.05	2	0.52	E1,E5 34.05	2	0.52	E1,E5 34.05	2	0.52	E1,E5 34.05	2	0.63	E1,E5 34.05	2	0.63	E1,E5 34.05	2	0.55	E1,E5,E9 59.37	3	1.23	E1,E5,E9 59.37	3	1.23	E1,E5,E9 59.37
	A7	0	0.66			0	0.50	0.00	0	1.04		0.00	0	0.54	E4 10.92	1	0.55	E4 10.92	1	0.54	E4 10.92	1	0.54	E4 10.92	1	0.52	E4,E7 24.67	2	0.65	E4,E7 24.67	2	0.81	E4,E7 24.67	2	1.12	E4,E7 24.67	2	1.12	E4,E7 24.67		
	A8	0	0.65			0	0.85	0.00	0	0.46		0.00	0	0.92		0.00	0	0.90		0.00	0	0.83		0.00	0	0.84		0.00	0	0.90		0.00	0	0.76		0.00	0	0.49		0.00	
	A9	0	0.60			0	0.88	0.00	0	0.46		0.00	0	0.91		0.00	0	0.89		0.00	0	0.83		0.00	0	0.80		0.00	0	0.89		0.00	0	0.74		0.00	0	0.55		0.00	
	A10	0	0.72			0	0.64	0.00	0	0.39		0.00	0	0.76		0.00	0	0.73		0.00	0	0.69		0.00	0	0.72		0.00	0	0.73		0.00	0	0.76		0.00	0	0.30	E10 6.53	0	0.30

**Legenda:**

Pior	Classificação	Melhor
------	---------------	--------

A1..A10 – Recursos; CT – Carga de Trabalho; E1..E10 – Eventos; t1..t20 – divisão temporal dos acontecimentos

L.E. – Lista de eventos atribuídos ao recurso, a cinzento quando foi atribuído um novo evento;

T.C. – Tempo chegada em minutos ao evento (a cinzento), tempo acumulado em minutos (a branco).

**Tabela 6.6 – Resultados das simulações com 10 recursos**





	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Total
<b>1<sup>a</sup> Simulação</b>	7,17	2,80	9,02	10,92	34,05	47,70	24,67	61,18	69,08	6,53	<b>273,12</b>
<b>2<sup>a</sup> Simulação</b>	7,17	2,80	9,02	10,92	34,05	8,38	24,67	15,95	59,37	6,53	<b>178,85</b>
<b>3<sup>a</sup> Simulação</b>	7,17	2,80	9,02	10,92	8,13	8,38	15,53	15,95	10,92	6,53	<b>95,35</b>
<b>4<sup>a</sup> Simulação</b>	7,17	2,80	9,02	10,92	8,13	8,38	15,53	15,95	13,23	6,53	<b>97,67</b>

**Tabela 6.8 – Comparação de tempos de chegada aos eventos por simulação com 10 recursos**

Na Tabela 6.6 e Tabela 6.7 estão resumidos os resultados das quatro simulações descritas anteriormente. Nestas tabelas, é possível consultar a atribuição dos eventos aos recursos pela sua ordem de chegada e mediante a sua classificação. Como se pode observar, a atribuição dos eventos variou entre as quatro simulações, sendo que, a maior variação encontra-se entre as duas primeiras.

É também nas duas primeiras simulações, sobretudo na primeira, que é possível encontrar um maior número de eventos atribuídos a um só recurso. Este acumular de eventos num só recurso fez com que o tempo de tratamento dos eventos aumenta-se bastante.

Na Tabela 6.8, é feita uma comparação dos tempos calculados para a chegada de um recurso ao evento. Analisando esta tabela é possível ver que é na terceira simulação onde se conseguiu melhores tempos de resposta aos eventos, tanto por evento, como no total de todos os eventos. A discrepância é maior ainda quando comparada com a primeira simulação, em que o valor total é quase o triplo da terceira.

Esta otimização dos tempos de resposta aos eventos, é conseguida através da introdução um maior número de elementos na classificação de recursos, visto que permite uma maior caracterização do estado atual do sistema e consequentemente uma escolha mais acertada do recurso a atribuir a um evento.

Relativamente à última simulação é possível constatar que o embora tenha sido usado mais um elemento na classificação de recursos, este não fez com que

fosse reduzido o tempo de resposta, pelo contrário, fez aumentar. Isto deve-se ao facto de que a prioridade do evento não pode ser vista como um fator de redução de tempo, mais sim como uma necessidade de priorizar os as ações dos recursos, independentemente de estas ações reduzam a eficiência do sistema.

### 6.2.2 Caso 2 – 5 Recursos, 10 Eventos

Com vista a testar o sistema num caso em existam menos recursos do que eventos ativos, foram feitas mais três simulações. Em termos de condições, estas simulações foram identificadas às anteriores, isto é, a 5ª Simulação é igual à 1ª Simulação, mas só com 5 recursos; a 6ª Simulação é igual à 2ª Simulação, mas só com 5 recursos; e a 7ª Simulação é igual à 3ª Simulação, mas só com 5 recursos. A 4ª Simulação não foi realizada com 5 recursos uma vez que o resultado não seria relevante para esta análise.

Os resultados das simulações estão ilustrados na Tabela 6.10. Na Tabela 6.9 é feita a comparação entre tempos de tratamento dos eventos.

Em relação aos resultados, em termos de comparação entre as novas simulações, constatou-se que existiu um comportamento semelhante as simulações anteriores, isto é, quantos mais elementos forem usados na classificação de recursos, menores são os tempos parciais e totais de tratamento dos eventos.

Comparando com os resultados das primeiras três simulações, verificou-se, como era de esperar, que os tempos de tratamento são superiores, uma vez que ao existirem menos recursos a carga de eventos é menos distribuída.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Total
<b>5ª Simulação</b>	13,73	38,20	9,02	8,00	52,15	8,38	15,53	52,15	60,05	20,78	<b>278,00</b>
<b>6ª Simulação</b>	13,73	38,20	9,02	8,00	22,50	8,38	15,53	15,32	52,82	36,45	<b>219,95</b>
<b>7ª Simulação</b>	13,73	9,67	7,37	8,00	17,05	7,37	13,73	17,05	21,92	9,67	<b>125,55</b>

Tabela 6.9 – Comparação de tempos de chegada aos eventos por simulação com 5 recursos



### 6.3 Testes em ambiente real

Num dos testes realizados em ambiente real, foram usados 5 recursos ligados ao sistema através de Agentes Mobile, representados no mapa por A1, A2, A3, A4 e A5, para tratamento de eventos. Admitiu-se que no início do teste já existiam 2 eventos ativos E1 e E2. O evento E1 estava atribuído ao Recurso A4 e o E2 ao A2. O recurso A2 já estava no local a tratar o evento enquanto que o recurso A4 estava a caminho do evento E1.

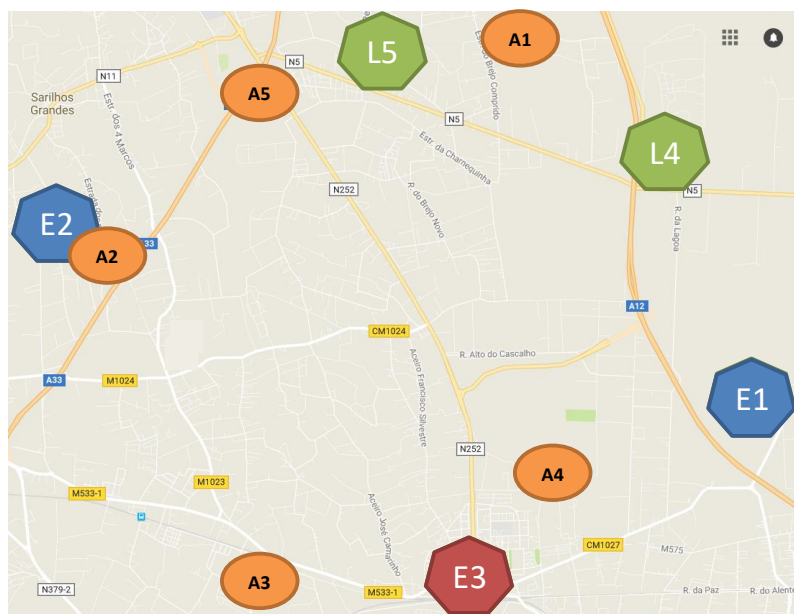


Figura 6.3 – Exemplo do mapa dos eventos

Usando como base este cenário inicial, foi provocado um novo evento, representado como E3 no mapa. Com acionamento da regra “Intrusão L3”, pela entrada de uma pessoa na área L3, e foi criado um novo Agente Ação que ficou responsável pela gestão das ações previstas para a área e por cada um dos estados. Depois de ter enviado o email previsto para o estado “Por Confirmar”, o evento foi atribuído<sup>18</sup> a um segurança (Agente Mobile) que confirmou o evento.

---

<sup>18</sup> Este processo de atribuição seguiu os passos descritos seguidamente para o estado “Confirmar”, mas para não estender a descrição, só serão descritos os passos do estado “Confirmar”.

Foram então enviados o email e sms previstos para o estado “Confirmado” e foi iniciado um novo processo para atribuição do evento, tendo em conta o estado atual do sistema: eventos ativos; recursos disponíveis; Localização do evento; Localização dos recursos.

A Tabela 6.11 resume os eventos ativos, descrevendo a sua localização, tipo e prioridade. A Tabela 6.12 descreve os recursos disponíveis, caracterizando-os em termos de localização, meio de locomoção, perfil, carga de trabalho, lista de eventos a tratar e estado.

	Evento 1	Evento 2	Evento 3
<b>Latitude</b>	38,64885	38,66601	38,63088
<b>Longitude</b>	-8,88107	-8,96553	-8,91523
<b>Prioridade</b>	Alta	Alta	Baixa
<b>Tipo</b>	Incêndio	Inundação	Intrusão

**Tabela 6.11 – Lista de eventos**

Usando como base esta informação o Agente de Ação procedeu à classificação dos recursos registados no sistema, obtendo assim a classificação destes mediante o seu perfil, carga de trabalho e tempo para chegar ao local.

	A1	A2	A3	A4	A5
<b>Latitude</b>	38,68316	38,66601	38,63195	38,64161	38,67807
<b>Longitude</b>	-8,91026	-8,96553	-8,94150	-8,90579	-8,94150
<b>Carga de trabalho</b>	0	1	0	1	0
<b>Lista de eventos</b>	-	Evento 2	-	Evento 1	-
<b>Estado</b>	Livre	A tratar evento	Livre	A caminho do evento	Livre
<b>Perfil</b>	Polícia	Segurança	Polícia	Polícia	Bombeiro
<b>Locomoção</b>	Mota	A pé	A pé	Carro	Carro

**Tabela 6.12 – Situação atual dos recursos**

A Tabela 6.13 resume a classificação final de cada um dos recursos<sup>19</sup>. Realçar que o A4 foi o único que recurso que foi necessário somar a distância da sua posição atual ao evento que tinha ainda por tratar, uma vez que este tinha prioridade sobre o novo evento.

	A1	A2	A3	A4	A5
<b>Perfil</b>	1	2	1	1	5
<b>Distância</b>	5,83	5,86	2,28	5,87	5,72
<b>Tempo de carro</b>	7,00			7,04	6,87
<b>Tempo a pé</b>	69,95	70,31	27,41	70,44	68,65
<b>Classificação</b>	0,31	70,31	27,41	0,37	1,03

**Tabela 6.13 – Cálculo para a classificação dos recursos**

Comparando os valores calculados, o Agente Ação atribui ao recuso A1 ao evento, visto que este obteve a melhor pontuação. Como o A1 não tinha nenhum evento atribuído, não foi necessária uma nova reatribuição para nenhum evento.

### 6.3.1 Resultados

Com a atribuição do evento E3 ao A1, foi possível comprovar o desempenho do sistema na atribuição de eventos, tendo em conta a localização do evento e da pessoa responsável pelo seu tratamento.

A execução deste teste demonstrou a rapidez esperada na atribuição de um evento a um recurso sem a necessidade de intervenção humana. Para além da rapidez na atribuição, o sistema conseguiu também garantir que escolheu o recurso disponível no sistema que irá realizar a tarefa num menor período de tempo.

---

<sup>19</sup> Neste teste, não foram usados os serviços do Google Maps, tendo sido usadas as formulas descritas anteriormente para cálculo da distância e tempo.

Comparando este processo com os tradicionais sistemas de vigilância, em que seria necessário ligar a cada um dos humanos envolvidos na monitorização do sistema para ver onde estavam e depois ligar ao mais próximo para lhe notificar que tinha de tratar o evento, constata-se uma significativa melhoria na rapidez de resposta ao evento. Esta melhoria torna-se ainda mais explícita num caso em que seja necessário fazer a reatribuição de eventos, em que o número de chamadas seria muito maior.

Outra das melhorias patentes nos testes efetuados, foi a redução da dependência dos humanos, uma vez que não foi necessária supervisão humana das câmaras para detetar um possível evento. Neste sistema, tal como descrito, os eventos foram detetados automaticamente, sem a necessidade de intervenção humana.

Na Figura 6.4 e na Figura 6.5 são ilustrados exemplos de deteções de incêndio e intrusões usando algoritmos no tratamento de vídeo.



Figura 6.4 – Exemplo de deteção de incêndio [93]



Figura 6.5 – Exemplo de deteção de intrusos e seguimento

A deteção de eventos baseada em algoritmos no tratamento de vídeo e/ou sensores, em substituição dos humanos em frente a um ecrã, pode aumentar o número de falsos positivos. No entanto, como neste sistema são usados diver-

sos tipos de sensores e é possível adequar as regras aos locais, foi possível comprovar, com os testes efetuados, que o sistema também permite uma redução significativa de falsos positivos. Por exemplo, num dos testes foi detetado que o algoritmo de deteção de incêndios não funcionava muito bem numa determinada hora do dia, visto que o sol ao bater na sala provocava um falso evento. Para eliminar este falso positivo, foi acrescentada à regra que detetava este evento, o parâmetro do sensor de incêndio, fazendo que a regra só fosse verdadeira quando os dois detetavam um incêndio.





## Conclusões e trabalhos futuros

Os sistemas de vigilância têm vindo a evoluir ao longo dos anos, porém, existem vários pontos onde ainda podem ser melhorados, como por exemplo, a redução da dependência dos humanos, a cooperação entre os sistemas e os humanos, a diminuição de falsos positivos na deteção de eventos, a utilização de vários tipos de sensores na deteção de eventos, a redução de custos de implementação e a descentralização.

Existem várias abordagens na literatura que visam contribuir para esta melhoria, no entanto, a grande maioria está focada na implementação de novos algoritmos para a deteção de eventos, identificação e seguimento de objetos e descentralização dos sistemas vigilância. Os artigos nesta área quase não abordam a temática da cooperação entre os sistemas e os humanos. Dos artigos estudados, nenhum visa a otimização dos sistemas de vigilância tendo em conta a localização dos eventos e dos humanos responsáveis pela segurança.

Tendo em conta esta lacuna, e tirando partido das novas tecnologias disponíveis hoje em dia no mercado, foi formulada a questão de investigação: **“Tendo em conta as novas tecnologias disponíveis atualmente, como se podem melhorar os atuais sistemas de vigilância?”**.

Para tentar responder a esta pergunta de investigação foi formulada a hipótese: **“Uma arquitetura de Sistemas Inteligentes de Vigilância, baseada em agentes de Software georreferenciados, pode aumentar a cooperação entre humanos e sensores, melhorando assim o desempenho global dos sistemas de vigilância atuais”**. Assim, e com base no estudo da arte feito no âmbito deste trabalho, é apresentada nesta dissertação uma nova proposta para arquiteturas de sistemas de vigilância, baseada em agentes de software inteligentes, georreferenciados, que cooperando com os humanos, alcança um comportamento global que permite aumentar a eficácia na resposta a eventos numa área vigiada [80], [81].

O sistema desenvolvido com base na arquitetura proposta, permitiu testar os conceitos propostos e validar os resultados previstos. O principal resultado obtido foi a **redução do tempo de resposta aos eventos**. Tirando partido das novas tecnologias e velocidades de comunicação disponíveis hoje em dia no mercado, o sistema baseado nesta arquitetura proposta, tem a capacidade de lidar com vários tipos de informações simultaneamente, como por exemplo: localização e prioridade dos eventos, localização e carga de trabalho dos recursos disponíveis. Assim, utilizando um novo método para atribuição de eventos a elementos de equipas de resposta, o sistema consegue reduzir os tempos de resposta a eventos.

Esta redução é feita à custa de algoritmos de atribuição de eventos, que para além da localização do evento e dos humanos, tem também conta outras variáveis, como a carga de trabalho e perfil do humano ou tipo e prioridade do evento. Com a conjunção destas variáveis, é possível uma atribuição de eventos significativamente mais eficiente, permitindo aumentar a eficiência e rapidez de resposta aos eventos.

Outro dos resultados obtidos, prende-se com a vantagem de o sistema permitir a **integração de múltiplas plataformas** (fixa e móvel), o que permite fornecer serviços de monitorização que contribuem para o aumento da segurança de pessoas e bens através de um novo tipo de serviço.

Também a utilização de uma arquitetura baseada em agentes facilita, por um lado, a **configurabilidade do sistema**, permitindo a integração de novos elementos e sem a necessidade de reconfiguração de todo o sistema, e por outro que **não existe um limite para a integração de novos elementos**. Para a integração de novos elementos, como, por exemplo, sensores, é necessário apenas a configuração do próprio sensor, ficando a partir daí disponível para ser usado em regras. Também a incorporação de um novo humano como Agente Mobile é facilitada com o uso desta arquitetura, visto que neste caso apenas é necessário definir um novo utilizador, com o respetivo perfil e áreas de atuação, passando este a estar disponível para tratamento de eventos.

Pode dizer-se assim que outro dos resultados deste sistema, é este ser composto por dispositivos que são *Plug-and-play*, permitindo-lhe ajustar-se às necessidades de cada situação específica, sem ter que voltar a configurar o sistema. Por outro lado, a utilização de algoritmos de complexidade linear no processamento de eventos, faz com que se possa concluir que o **sistema é escalável**.

Esta nova abordagem aos sistemas de vigilância permite uma tipologia adaptável, facilitando o **uso de diversos tipos de sensores** na deteção de eventos, permitindo inferir um maior número e tipos de eventos. Por outro lado, a combinação destas fontes de informação, permite **reduzir o número de falsos positivos**, uma vez que a “leitura errada” de um sensor pode ser sempre confirmada com o auxílio de outro. O aumento da quantidade de informações processadas permite o aumento da decisão de qualidade, reduzindo também os custos associados a situações de falso alarme.

A adoção desta arquitetura multiagente elimina a dependência de uma unidade central de processamento e possibilita o **processamento distribuído**. Esta capacidade de processamento distribuído (que pode usar até mesmo dispositivos de baixo custo, tais como Raspberry PI) e localizada (com o uso de agentes de software) permite que as ações sejam desencadeadas, com base em regras e na posição geográfica dos eventos e recursos.

Os resultados obtidos, permitem assegurar que a arquitetura proposta irá: reduzir os custos de supervisão de instalações (utilizando unidades de processamento mais económicas); minimizar custos com pessoal de vigilância (menos dependência dos seres humanos na monitorização e deteção); aumentar rapidez de reação de resposta aos eventos (atribuindo recursos tendo em conta a sua localização); diminuir falsos eventos (utilizando diversos tipos de sensores na deteção de eventos); minorar a dependência de um ponto central (utilizando uma solução distribuída).

Estes resultados foram comprovados com o auxílio de simulações e testes efetuados com um protótipo operacional, que permitiu a validação da arquitetura face aos objetivos inicialmente definidos. Algumas destas simulações e testes são descritos em pormenor no capítulo anterior, onde é demonstrada a rapidez e eficácia de resposta do sistema aos eventos, tendo em conta fatores como a distância do recurso ao evento, carga de trabalho e perfil do recurso e prioridade dos eventos.

É de salientar também que esta arquitetura foi usada como base do sistema de vigilância desenvolvido no projeto QREN DVA, permitindo a criação de um produto que compete com os sistemas tradicionais.

Na fase final do trabalho desenvolvido foram feitos desenvolvimentos adicionais que permitiram integrar no sistema desenvolvido os resultados de outro projeto de investigação, o ServRobot, mencionado anteriormente. Estas modificações permitiram a integração de robots móveis como agentes do sistema, quer a nível de disponibilização de dados sensoriais, quer para validação *in loco* de eventos.

## **7.1 Trabalho Futuro**

Em termos de trabalho futuro, foi efetuado um esforço para a identificação de pontos onde possa ser melhorada a arquitetura proposta. Um dos pontos já identificados é a possibilidade de desativação (ou sugestão de desativação) de regras que originem sucessivos eventos cancelados. Neste caso, podem ser de-

finidas regras que permitam a desativação de regras com um número elevado de confirmações negativas.

Outro exemplo onde este sistema pode ser aperfeiçoado está relacionado com capacidade deste em fazer prognóstico de eventos. Isto é, pode ser integrada no sistema uma ferramenta de *Data Mining*, que através da análise dos parâmetros, consiga detetar alterações de valores que levem a inferir a futura ocorrência de um evento.

Outro aspeto que pode ser estudado é capacidade de distribuição da tomada de decisão na atribuição de eventos. Isto é, na arquitetura proposta a tomada de decisão é feita por um Agente Ação criado automaticamente para cada um evento dos eventos e que mediante a recolha de informação de todo o sistema classifica os recursos e escolhe um. Como trabalho futuro, pode ser estudada a hipótese de ao ser criado um evento seja o Agente Mobile/Interface a dizer que fica com esse evento em vez de espera que o Agente Ação reúna toda a informação.

A aplicação desta arquitetura a outro tipo de cenários, como a monitorização de comboios, pode também ser um dos alvos de trabalho futuro. Neste caso, o sistema pode ser usado para monitorizar as várias variáveis disponíveis no comboio. A monitorização e fusão da informação recolhida no comboio pode permitir a implementação de regras auxiliem na prevenção de avarias ou mesmo de acidentes [76].

Por último, pode também ser desenvolvido um modelo semântico de apoio às regras assim como um modelo de dados para os parâmetros, eventos, ações e tarefas.



## Referências

- [1] M. Valera and S. Velastin, "Intelligent distributed surveillance systems: a review," *IEEE Proc. - Vision, Image, Signal Process.*, vol. 152, no. 2, p. 192, 2005.
- [2] P. D. J. M. Leimeister, "Collective Intelligence," *Bus. Inf. Syst. Eng.*, vol. 2, no. 4, pp. 245–248, 2010.
- [3] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognit. Lett.*, vol. 34, no. 1, pp. 3–19, Jan. 2013.
- [4] J. Albusac, D. Vallejo, J. J. Castro-Schez, P. Remagnino, C. Gonzalez, and L. Jimenez, "Monitoring Complex Environments Using a Knowledge-Driven Approach Based on Intelligent Agents," *IEEE Intell. Syst.*, vol. 25, no. 3, pp. 24–31, May 2010.
- [5] P. Korshunov and W. T. Ooi, "Critical video quality for distributed automated video surveillance," *Proc. 13th Annu. ACM Int. Conf. Multimed. - Multimed. '05*, p. 151, 2005.
- [6] S. Onofre, J. Pimentão, and P. Sousa, "Georeferenced Dynamic Event Handling," in *Technological Innovation for Cloud-Based Engineering Systems SE - 27*, vol. 450, L. M. Camarinha-Matos, T. A. Baldissera, G. Di Orio, and F. Marques, Eds. Springer International Publishing, 2015, pp. 251–258.
- [7] D. Lyon, *Surveillance studies: An overview*. Polity, 2007.
- [8] H. Sean and G. Joshua, *The Surveillance Studies Reader*. 2007.
- [9] S. L. Nock and W. G. Staples, *The Culture of Surveillance: Discipline and*

*Social Control in the United States*, vol. 27. 1998.

- [10] World Health Organization, *Food and nutrition surveillance systems Food and nutrition*. 2014.
- [11] M. D. Salman, *Ticks and tick-borne diseases: geographical distribution and control strategies in the Euro-Asia region*. Wallingford: CABI, 2013.
- [12] T. Nagalakshmi, "A Study on usage of CCTV Surveillance system with spcial reference to business outlets in hyderabad," *Tactful Manag. Res. J.*, vol. 1, no. 2, pp. 1–12, 2012.
- [13] H. Kruegle, *CCTV Surveillance: Video practices and technology*. Butterworth-Heinemann, 2011.
- [14] T. D. Rätty, "Survey on contemporary remote surveillance systems for public safety," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 40, no. 5, pp. 493–515, 2010.
- [15] "Airport Surface Detection Equipment, Model X (ASDE-X). Sensis,," 2010. [Online]. Available: <http://www.sensis.com/docs/128/>. [Accessed: 20-Sep-2010].
- [16] M. E. Weber and M. L. Stone, "Low altitude wind shear detection using airport surveillance radars," in *Proceedings of 1994 IEEE National Radar Conference*, 1995, pp. 52–57.
- [17] A. Pozzobon, G. Sciutto, and V. Recagno, "Security in Ports: the User Requirements for Surveillance System," in *Advanced Video-Based Surveillance Systems*, vol. 488, C. S. Regazzoni, G. Fabri, and G. Vernazza, Eds. Boston, MA: Springer US, 1999, pp. 18–26.
- [18] C. Nwagboso, *User focused Surveillance Systems Integration for Intelligent Transport Systems*. Boston, MA: Springer US, 1999.
- [19] B. P. Lai Lo, J. Sun, S. A. Velastin, and J. A. V. S.A. Lo, Benny Ping Lai , Sun, "Fusing Visual and Audio Information in a Distributed Intelligent Surveillance System for Public Transport Systems," in *Acta Automatica Sinica*, vol. 29, no. 3, 2003, pp. 393–407.
- [20] B. Abreu *et al.*, "Video-based multi-agent traffic surveillance system," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, 2000, pp. 457–462.
- [21] T. D’Orazio, C. Guaragnella, and T. D. and C. Guaragnella, "A Survey of Automatic Event Detection in Multi-Camera Third Generation Surveillance Systems," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 1, p. 1555001, 2015.



- [22] D. C. Gray, *The Fourth Amendment in an Age of Surveillance*. 2017.
- [23] R. Bergquist, G.-J. Yang, S. Knopp, J. Utzinger, and M. Tanner, "Surveillance and response: Tools and approaches for the elimination stage of neglected tropical diseases," *Acta Trop.*, vol. 141, pp. 229–234, 2015.
- [24] C. Calba *et al.*, "Surveillance systems evaluation: a systematic review of the existing approaches," *BMC Public Health*, vol. 15, no. 1, p. 448, 2015.
- [25] C. T. Lewis and C. Short, *A Latin Dictionary*. 1894.
- [26] A. Backlund, "The definition of system," *Kybernetes*, vol. 29, no. 4, pp. 444–451, Jun. 2000.
- [27] D. M. Buede, *The Engineering Design of Systems*, Second Edi. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.
- [28] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2/E. 2007.
- [29] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, vol. 4. 2012.
- [30] T. Saridakis, "Design Patterns for Graceful Degradation," 2009, pp. 67–93.
- [31] P. Veríssimo and L. Rodrigues, *Distributed Systems for System Architects*, vol. 1. Boston, MA, MA: Springer US, 2001.
- [32] "CORBA.ORG." [Online]. Available: <http://www.corba.org/>.
- [33] "The Open Group." [Online]. Available: <http://www.opengroup.org/>.
- [34] N. R. Jennings, "On agent-based software engineering," *Artif. Intell.*, vol. 117, no. 2, pp. 277–296, 2000.
- [35] P. Sousa, "Um enquadramento para a catalogação automática de dados. Uma abordagem Multiagentes," Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, 2004.
- [36] Q. Mahmoud, "Software Agents: Characteristics and Classification," *Cis.Uoguelph.Ca*.
- [37] M. Schumacher, *Objective Coordination in Multi-Agent System Engineering*, vol. 2039. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
- [38] H. S. Nwana, "Software Agents : An Overview," *Knowl. Eng. Rev.*, vol. 11, no. 3, pp. 205–244, 1996.
- [39] M. R. Genesereth and S. P. Ketchpel, "Software Agents," *Commun ACM*. pp. 1–15, 1994.

- [40] S. Poslad, "Specifying protocols for multi-agent systems interaction," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, no. 4, p. 15–es, 2007.
- [41] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," *Softw. Agents. MIT Press. Cambridge*, vol. 239, pp. 1–22, 1995.
- [42] C. Bădică, Z. Budimac, H. D. Burkhard, and M. Ivanović, "Software agents: Languages, tools, platforms," *Comput. Sci. Inf. Syst.*, vol. 8, no. 2, pp. 255–296, 2011.
- [43] "Special issue on video communications, processing, and understanding for third generation surveillance systems," *Proc. IEEE*, vol. 89, no. 10, pp. 1355–1539, 2001.
- [44] Y.-C. T. Chung-Shuo Fan, Jia-Ming Liang, Yi-Ting Lin, Kun-Ru Wu, Kuan-Yi Li, Ting-Yu Lin, "A Survey of Intelligent Video Surveillance Systems: History, Applications and Future," in *Volume 274: Intelligent Systems and Applications*, 2015, pp. 1479–1488.
- [45] D. Ballard and G. Stockman, *Computer Vision*, vol. 253, no. Sept. 1982.
- [46] G. Rodríguez-Canosa, J. del Cerro Giner, and A. Cruz, "Detection and Tracking of Dynamic Objects by Using a Multirobot System: Application to Critical Infrastructures Surveillance," *Sensors*, vol. 14, no. 2, pp. 2911–2943, 2014.
- [47] H. Zhang, W. Gao, X. Chen, and D. Zhao, "Object detection using spatial histogram features," *Image Vis. Comput.*, vol. 24, no. 4, pp. 327–341, Apr. 2006.
- [48] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, and K.-R. Müller, "From outliers to prototypes: Ordering data," *Neurocomputing*, vol. 69, no. 13–15, pp. 1608–1618, Aug. 2006.
- [49] R. T. Collins *et al.*, "A system for video surveillance and monitoring," Aug. 2000.
- [50] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," *Comput. Vision – ECCV 2002*, vol. 2353, pp. 343–357, 2002.
- [51] V. Nair and J. J. Clark, "Automated Visual Surveillance Using Hidden Markov Models," pp. 1–5, 2002.
- [52] W. Niu, J. Long, D. Han, Y. Wang, and S. Barbara, "Human Activity Detection and Recognition for Video Surveillance," pp. 1–4, 2004.
- [53] Z. Zhi-Hong, "Lane detection and car tracking on the highway," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 450–456, 2003.

- [54] H. W.-M. LOU Jian-Guang, LIU Qi-Feng, TAN Tie-Niu, "3-D Model Based Visual Traffic Surveillance," *Acta Autom. Sin.*, vol. 29(03), pp. 434–449, 2003.
- [55] J. M. Ferryman, S. J. Maybank, and A. D. Worrall, "Visual surveillance for moving vehicles," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 187–197, 2000.
- [56] C. Decleir, M.-S. Hacid, and J. Kouloumdjian, "A database approach for modeling and querying video data," in *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, 1999, pp. 6–13.
- [57] E. Stringa and C. S. Regazzoni, "Content-based retrieval and real time detection from video sequences acquired by surveillance systems," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, 1998, vol. 3, pp. 138–142.
- [58] S. A. S. Velastin, L. Khoudour, B. P. L. Lo, J. Sun, and M. A. Vicencio-Silva, "PRISMATICA: a multi-sensor surveillance system for public transport networks," in *12th IEE International Conference on Road Transport Information & Control - RTIC 2004*, 2004, vol. 2004, no. 501, pp. 19–25.
- [59] N. Siebel and S. Maybank, "The advisor visual surveillance system," *ECCV 2004 Work. Appl.*, 2004.
- [60] F. Segor, A. Bürkle, and S. Müller, "Architecture of a Security and Surveillance System," ... *Syst.*, no. c, pp. 20–24, 2012.
- [61] M. K. Lim, S. Tang, and C. S. Chan, "ISurveillance: Intelligent framework for multiple events detection in surveillance videos," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4704–4715, Aug. 2014.
- [62] D. Vallejo, J. Albusac, J. J. Castro-Schez, C. Glez-Morcillo, and L. Jiménez, "A multi-agent architecture for supporting distributed normality-based intelligent surveillance," *Eng. Appl. Artif. Intell.*, vol. 24, no. 2, pp. 325–340, Mar. 2011.
- [63] J. Tejedor, M. Patricio, and J. Molina, "Multi-agent Based Distributed Semi-automatic Sensors Surveillance System Architecture," in *Distributed Computing and Artificial Intelligence*, vol. 79, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 317–324.
- [64] M. a. Patricio, J. Carbó, O. Pérez, J. García, and J. M. Molina, "Multi-Agent Framework in Visual Sensor Networks," *EURASIP J. Adv. Signal Process.*, vol. 2007, pp. 1–22, 2007.
- [65] Z. A. Baig, "Multi-agent systems for protecting critical infrastructures: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 1151–1161, May 2012.

- [66] R. Abielmona, E. M. Petriu, and T. Whalen, "Multi-Agent System Information Fusion for Environment Monitoring," pp. 1774–1779, 2006.
- [67] A. N. Belbachir, *Smart Cameras*. Boston, MA: Springer US, 2010.
- [68] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed Embedded Smart Cameras for Surveillance Applications," *Computer (Long. Beach. Calif.)*, vol. 39, no. 2, pp. 68–75, Feb. 2006.
- [69] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001.
- [70] M. E.- Boudihir and K. A. Al-Shalfan, "Intelligent video surveillance system architecture for abnormal activity detection," in *The International Conference on Informatics and Applications (ICIA2012)*, 2012, pp. 102–111.
- [71] J. L. L. Castro, M. Delgado, J. Medina, and M. D. D. Ruiz-Lozano, "Intelligent surveillance system with integration of heterogeneous information for intrusion detection," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11182–11192, Sep. 2011.
- [72] D. Vallejo, L. M. García-Muñoz, J. Albusac, C. Glez-Morcillo, L. Jiménez, and J. J. Castro-Schez, "Developing Intelligent Surveillance Systems with an Agent Platform," in *Agent and Multi-Agent Systems. Technologies and Applications: 6th KES International Conference, KES-AMSTA 2012, Dubrovnik, Croatia, June 25-27, 2012. Proceedings*, G. Jezic, M. Kusek, N.-T. Nguyen, R. J. Howlett, and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 199–208.
- [73] D. Vallejo, F. J. Villanueva, L. M. García, C. Gonzalez, and J. Albusac, "An agent-based approach to understand events in surveillance environments," *Proc. - 2013 IEEE/WIC/ACM Int. Jt. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IATW 2013*, vol. 3, pp. 100–103, 2013.
- [74] J. Albusac, D. Vallejo, J. J. J. Castro-Schez, C. Glez-Morcillo, and L. Jiménez, "Dynamic weighted aggregation for normality analysis in intelligent surveillance systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2008–2022, Mar. 2014.
- [75] L. M. Camarinha-matos, "SCIENTIFIC RESEARCH Unit 2 : SCIENTIFIC METHOD," 2009.
- [76] S. Onofre, J. Gomes, J. P. Pimentão, and P. A. Sousa, "Data Fusion of Georeferenced Events for Detection of Hazardous Areas," 2017, pp. 81–89.

- [77] T. Duarte, J. P. Pimentao, P. Sousa, and S. Onofre, "Biometric Access Control Systems: A Review Technologies to Improve Their Efficiency," *2016 IEEE Int. Power Electron. Motion Control Conf.*, pp. 795–800, 2016.
- [78] S. Onofre, P. M. Silvestre, J. P. Pimentao, and P. Sousa, "Surpassing bluetooth low energy limitations on distance determination," *Proc. - 2016 IEEE Int. Power Electron. Motion Control Conf. PEMC 2016*, no. 1, pp. 843–847, 2016.
- [79] S. Onofre, B. Caseiro, J. P. Pimentão, and P. Sousa, "Using Fuzzy Logic to improve BLE indoor positioning system," in *Technological Innovation for Cyber-Physical Systems*, M. L. Camarinha-Matos, A. J. Falcão, N. Vafaei, and S. Najdi, Eds. Cham: Springer International Publishing, 2016, pp. 169–177.
- [80] S. Onofre, P. Sousa, and J. P. Pimentao, "Multi-sensor geo-referenced surveillance system," in *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, 2015, pp. 1–6.
- [81] S. Onofre, P. Sousa, and J. P. Pimentao, "Geo-referenced multi-agent architecture for surveillance," *Power Electronics and Motion Control Conference and Exposition (PEMC), 2014 16th International*. pp. 455–460, 2014.
- [82] J. Claro, B. Dias, B. Rodrigues, J. P. Pimentao, P. Sousa, and S. Onofre, "Autonomous robot integration in surveillance system: Architecture and communication protocol for systems cooperation," *Power Electronics and Motion Control Conference and Exposition (PEMC), 2014 16th International*. pp. 713–719, 2014.
- [83] B. Dias, B. Rodrigues, J. Claro, J. P. Pimentão, P. Sousa, and S. Onofre, "Architecture and message protocol proposal for robot's integration in multi-agent surveillance system," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8536 LNAI, pp. 366–373.
- [84] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [85] H. Huang, D. Zhu, and F. Ding, "Dynamic Task Assignment and Path Planning for Multi-AUV System in Variable Ocean Current Environment," *J. Intell. Robot. Syst.*, Aug. 2013.
- [86] S. Sariel, T. Balch, and N. Erdogan, "Naval Mine Countermeasure Missions," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 45–52, Mar. 2008.
- [87] Anmin Zhu, S. X. S. X. Yang, A. Zhu, and S. X. S. X. Yang, "A neural

- network approach to dynamic task assignment of multirobots,” *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1278–87, Sep. 2006.
- [88] B. Bethke, M. Valenti, and J. How, “UAV Task Assignment,” *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 39–44, Mar. 2008.
  - [89] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, “Distributed multi-robot task assignment and formation control,” in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 128–133.
  - [90] B. J. Moore and K. M. Passino, “Distributed Task Assignment for Mobile Agents,” *IEEE Trans. Automat. Contr.*, vol. 52, no. 4, pp. 749–753, Apr. 2007.
  - [91] J. Mclurkin and D. Yamins, “Dynamic Task Assignment in Robot Swarms,” *Robot. Sci. Syst.*, 2005.
  - [92] R. W. Sinnott, “Virtues of the Haversine,” *Sky Telescope*, vol. 68, no. 2, p. 159, 1984.
  - [93] P. Santana, P. Gomes, and J. Barata, “A vision-based system for early fire detection,” in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2012, pp. 739–744.
  - [94] H. T. Valentim, “Um estudo semântico-enunciativo de predicados subjectivos do português,” Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa, 2004.
  - [95] R. A. Finkel and J. L. Bentley, “Quad trees a data structure for retrieval on composite keys,” *Acta Inform.*, vol. 4, no. 1, pp. 1–9, 1974.